

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

CONTROLE DE SISTEMAS NÃO LINEARES ATRAVÉS DE REDES NEURAIS

Este exemplar corresponde à redação final da tese
defendida por Edilberto Pereira Teixeira

e aprovada pela Comissão
Julgadora em 30 / 7 / 1991.

Orientador Fernando A.C. Gomide

Por : Edilberto Pereira Teixeira

Orientador : Prof. Fernando A.C. Gomide †

Co-orientador : Prof. Kenneth Loparo n.t.c

Tese apresentada à Faculdade de Engenharia
Elétrica, da Universidade Estadual de
Campinas, como parte dos requisitos exigidos
para obtenção do Título de Doutor em
Engenharia Elétrica.

Julho de 1991

Bc/19109436

AGRADECIMENTOS

Agradecemos ao Prof. Fernando A. C. Gomide pela orientação e apoio dedicados tanto durante a fase inicial dos trabalhos na Universidade Estadual de Campinas, como no período de desenvolvimento na *Case Western Reserve University*.

Ao Prof. Kenneth Loparo agradecemos pela co-orientação e colaboração prestada na elaboração deste trabalho, durante o período em que foi realizado na *Case Western Reserve University, Systems Engineering Department*. Este apoio se estendeu desde a discussão das idéias básicas até a versão em língua inglesa.

Contou-se com apoio financeiro da CAPES, do CNPq/RHAE e da Universidade Federal de Uberlândia.

Agradecemos a *Case Western Reserve University, Systems Engineering Department* e o Laboratório de Computação e Automação do Departamento de Engenharia de Computação e Automação Industrial da Faculdade de Engenharia Elétrica da Unicamp pela utilização do equipamentos.

A Luci, Raquel, Adriano, Cristiane e Flavia,

dedico este trabalho.

Control of Nonlinear Systems Using Neural Networks

ABSTRACT

This thesis presents an approach for controlling a nonlinear system using an adaptive scheme implemented through neural networks. For a class of systems, necessary conditions for teaching the system input-output mapping to a multi-layer neural network trained by the generalized delta rule are presented. A feedback linearization procedure, using feedforward neural networks, is also introduced in this work. The method can be applied for unknown continuous time systems as well as discrete time or sampled systems. The feedback linearization procedure can be either adaptively accomplished, having the neural network weights adjusted during normal operation of the plant, or in a non adaptive mode, having the weights adjusted off-line.

In the proposed scheme, two neural networks are trained in two different stages. In the first stage, the nonlinear system is excited several times to teach the inverse dynamics of the system to a neural network. In the second stage, the system is again excited several times to train a second neural network with input signals that will control the nonlinear system in the desired way.

Controle de Sistemas Não Lineares Através de Redes Neurais

Sumário

Esta tese apresenta um método para se controlar sistemas não lineares por meio de um esquema adaptativo implementado através de rede neurais artificiais. Para uma classe específica de sistemas, são apresentadas as condições necessárias para se ensinar o mapeamento de entrada-saída a uma rede neural treinada pela regra delta generalizada. Apresenta-se também neste trabalho, um método de linearização por realimentação, usando-se redes neurais. O método pode ser aplicado para sistemas não identificados, de tempo contínuo, discreto ou sistemas amostrados. Os procedimentos para linearização por realimentação podem ser realizados off-line ou de forma adaptativa, onde os pesos da rede neural são ajustados durante a operação normal do sistema.

No esquema proposto, duas redes neurais são ajustadas em dois estágios diferentes. No primeiro estágio, o sistema não linear é excitado várias vezes para ensinar a dinâmica inversa do sistema a uma rede neural. No segundo estágio, o sistema é novamente excitado para treinar uma segunda rede neural com os sinais que controlam o sistema na forma desejada.

INDICE

Controle de Sistemas Não Lineares Por Meio de Redes Neurais

Introdução	0-1
CAP. I - Projeto de redes neurais de múltiplas camadas para identificação precisa de mapeamentos não lineares	I-1
1. Determinação do número de amostras	I-2
2. Definição da estrutura de redes neurais de múltiplas camadas	I-8
3. Procedimentos para identificação de mapeamentos não-lineares usando-se redes neurais	I-11
4. Treinamento on-line de redes neurais	I-12
5. Conclusões	I-14
CAP. II - Linearização por realimentação em sistemas dinâmicos não lineares	II-1
1. Linearização por realimentação em sistemas SISO de tempo contínuo não-lineares	II-10
1.1 Conceito de grau-relativo	II-11
1.2 Linearização por realimentação do modelo de entrada-saída	II-13
1.3 Linearização exata por realimentação no espaço de estado	II-15

1.4 Dinâmica nula	II-22
2. Linearização por realimentação de sistemas de tempo contínuo de múltiplas entradas e múltiplas saídas (MIMO)	II-24
3. Linearização por realimentação de sistemas de tempo discreto do tipo SISO	II-27
3.1 Definições	II-28
3.2 Linearização por realimentação em sistemas de tempo discreto SISO tipo Σ_N	II-33
3.3 Linearização por realimentação em sistemas não lineares SISO de tempo discreto da classe Σ	II-37
3.4 Linearização por realimentação em sistemas MIMO de tempo discretos	II-39
4. Efeitos do processo de amostragem em linearização por realimentação	II-40
5. Conclusões	II-43
 CAP. III - Dinâmica inversa de sistemas não lineares usando redes neurais	 III-1
1. Método de inversão de Hirschorn	III-2
2. Dinâmica inversa de sistemas contínuos não lineares usando-se redes neurais.	III-10
2.1 Procedimento para identificação da dinâmica inversa usando-se redes neurais	III-13
2.2 Obtenção do sistema inverso aproximado usando-se valores de saída amostrados	III-15
3. Dinâmica inversa de sistemas de tempo discreto usando redes neurais	III-29
4. Conclusão	III-35

CAP. IV - Linearização por realimentação usando redes neurais	IV-1
1. Linearização por realimentação em sistemas não lineares de tempo discreto através do uso de redes neurais	IV-2
1.1 Linearização de entrada-saída por realimentação em sistemas de tempo discreto	IV-3
1.2 Linearização por realimentação no espaço de estados para sistemas de tempo discreto.	IV-13
2. Linearização por realimentação através do uso de redes neurais para de sistemas de tempo contínuo não identificados	IV-22
2.1 Linearização por realimentação de sistemas de tempo contínuo do tipo ψ_N	IV-23
2.2 Linearização por realimentação de sistemas de tempo contínuo do tipo ψ_s (sem acesso ao estado)	IV-30
3. Conclusão	IV-48
Conclusão Geral	C-1
Apêndice I - Redes neurais artificiais de várias camadas	Ap. I
Apêndice II - Sistemas não lineares - conceitos básicos notação usada	Ap. II
Bibliografia	B-1

CONTROLE DE SISTEMAS NÃO LINEARES POR MEIO DE REDES NEURAIS

INTRODUÇÃO

Nos últimos anos, tem havido um considerável progresso nas técnicas de controle de sistemas dinâmicos não lineares. Importantes questões tais como controlabilidade, estabilidade, desacoplamento de entrada-saída e desacoplamento de distúrbios têm sido analisadas por vários pesquisadores, obtendo-se resultados animadores. Entretanto, no que se refere ao problema de identificação de sistemas não lineares, poucos resultados práticos com soluções viáveis foram obtidos.

Sem dúvida, a área de redes neurais se constitui em uma direção bastante animadora para a solução de problemas de controle de sistemas não lineares e não identificados. Assim sendo, este trabalho trata especificamente deste assunto e apresenta soluções viáveis, resolvendo-se o problema de controle através de linearização por realimentação e com uso de redes neurais.

Pode-se destacar os seguintes trabalhos desenvolvidos independentemente: Li(1989), Narendra(1990), Nguyen(1990), Chen(1990) e Loparo & Teixeira(1990) e Teixeira(1991b). Nestes trabalhos, redes neurais foram empregadas na solução de diferentes problemas de controle e identificação de sistemas não lineares. O trabalho de Narendra trata principalmente da identificação de sistemas não lineares através de redes neurais. Chen identifica uma classe de sistemas não lineares de tempo discreto para a solução do problema de trajetória. No presente trabalho, a dinâmica inversa de sistemas não lineares é identificada usando-se uma rede neural de uma ou mais camadas intermediárias. Uma outra rede neural, de características similares, é usada para assimilar

a função de linearização por realimentação. O método pode ser aplicado a uma classe mais ampla de sistemas não lineares, visto que tanto sistemas de tempo discreto como contínuo podem ser identificados e linearizados por realimentação. O método pode ser também aplicado para sistemas amostrados e relaxa a restrição de que somente sistemas da classe lineares-analíticos poderiam ser linearizados por realimentação (Isidori, 1989).

Para sistemas de tempo discreto, ou sistemas contínuos amostrados, é apresentada uma extensão adaptativa do método que permite treinamento on-line das redes neurais, quando as variações dos parâmetros ou do modelo forem relativamente lentas.

No capítulo I, trata-se do problema de dimensionamento de redes neurais de camadas múltiplas e se estabelecem condições suficientes para a determinação do número de elementos nas camadas intermediárias, para emprego em aproximação funcional. Redes neurais de várias camadas tem sido usadas em diversas aplicações tais como reconhecimento de padrões e identificação de mapeamentos não lineares através de aproximação funcional. O problema de se definir arquiteturas de redes neurais tem sido investigado por vários autores, veja por exemplo Wang (1989), Funahashi (1989), Baum (1989) e Teixeira (1990,1991). No capítulo I este problema é direcionado às soluções específicas de sistemas de controle não lineares.

No capítulo II, faz-se uma revisão do método analítico de linearização por realimentação em sistemas de tempo discreto e contínuo. Apresenta-se uma análise dos vários procedimentos envolvendo linearização por realimentação, além de adicionar considerações relacionadas ao objetivo principal deste trabalho, isto é, linearização por realimentação usando-se redes neurais. São apresentados conceitos básicos tais como, grau relativo e

dinâmica nula para sistemas não lineares de tempo discreto e contínuo, SISO e MIMO. Estes conceitos formam a base teórica das técnicas de linearização por realimentação e são, portanto, necessários para uma perfeita compreensão das vantagens e limitações do método, aqui proposto.

No capítulo III, trata-se do assunto de inversão de sistemas dinâmicos não lineares. O capítulo é iniciado com uma revisão dos métodos analíticos de inversão e em seguida é introduzido um método que utiliza redes neurais para identificar a dinâmica inversa do sistema. Através do método proposto, identifica-se o sinal de excitação capaz de produzir a saída atual, baseado nas condições anteriores do sistema. O método pode ser aplicado tanto para sistemas de tempo discreto como sistemas de tempo contínuo ou amostrado.

No capítulo IV, propõe-se então um método para se controlar sistemas não lineares através de um esquema adaptativo com uso de redes neurais. Para um classe específica de sistemas não lineares, são estabelecidos esquemas de treinamento das redes neurais, e são também apresentados resultados de simulação onde sistemas não lineares são linearizados por realimentação. O método proposto pode ser aplicado tanto a sistemas de tempo discreto como contínuo ou amostrado. O treinamento das redes pode ser feito de forma adaptativa, sendo os pesos das redes neurais ajustados on-line, ou de forma não adaptativa. Os métodos apresentados são ilustrados através de exemplos simulados em computador digital.

No Apêndice I, são apresentados os fundamentos teóricos de redes neurais de várias camadas, mostrando-se as suas principais aplicações e enfatizando o seu emprego em aproximação funcional. O Apêndice II apresenta os principais conceitos de sistemas não lineares usados neste trabalho.

As pesquisas foram, em parte, realizados no Departamento de Engenharia de Sistemas da Case Western Reserve University, com coordenação do Prof. Kenneth Loparo, como parte do programa de Doutorado sendo realizado no Departamento de Computação e Automação da Universidade Estadual de Campinas, sob orientação do Prof. Fernando A. C. Gomide e com suporte da Capes, CNPQ e Universidade Federal de Uberlândia.

CAPITULO I

PROJETO DE REDES NEURAS DE MÚLTIPLAS CAMADAS PARA IDENTIFICAÇÃO PRECISA DE MAPEAMENTOS NÃO LINEARES.

Redes neurais de várias camadas têm sido usadas em diversas aplicações tais como reconhecimento de padrões e identificação de mapeamentos não lineares. Em Rumelhart (1987), Lippmann (1987) e Apêndice I, se encontram descrições de tais redes e suas aplicações. Teoricamente, a aplicação de redes neurais para aproximação funcional pode ser analisada usando-se o teorema de Wierstrass, Haaser (1971). Basicamente, este teorema afirma que qualquer função contínua de valores reais, definida em um intervalo limitado, pode ser aproximada por um polinômio. Se a função de ativação de cada elemento da rede neural for uma função contínua de valor real, ela também poderá ser aproximada por um polinômio e conseqüentemente a relação funcional de entrada-saída da rede poderá também ser aproximada por um polinômio. Assim sendo será sempre possível definir uma rede neural de múltiplas camadas para atuar como aproximação de um mapeamento não linear específico.

O estabelecimento de uma estrutura de rede neural deve considerar tanto o número de amostras usadas na fase de treinamento, como a precisão desejada nas previsões a serem feitas fora do conjunto usado para treinamento. Este capítulo apresenta um estudo da definição do número amostras de treinamento, e introduz algumas proposições para determinação do número de elementos em cada camada intermediária da rede.

O problema de se definir arquiteturas de redes neurais tem sido investigado por vários autores, veja por exemplo

Wang (1989), Funahashi (1989)) e Teixeira (1990,1991). As seguintes causas de imprecisão devem ser consideradas no emprego de redes neurais para aproximação de mapeamentos não lineares:

a- o número de amostras, usadas para treinamento, pode ser insuficiente para captar a estrutura real da função;

b- a rede neural poderá estar subdimensionada, não possuindo número suficiente de elementos na camada intermediária e portanto, a tolerância desejada não poderá ser atingida.

O item (a), pode ser considerado como um problema de projeto da rede, e para isso, torna-se necessário um certo conhecimento do mapeamento a ser ensinado à rede. Este problema é analisado na próxima seção, onde se propõe um procedimento prático para se determinar o número suficiente de amostras. A segunda fonte de imprecisão pode ser evitada pela escolha correta da arquitetura da rede neural. Para isso, são estabelecidos limites para o número de elementos das camadas intermediárias.

1. Determinação do número de amostras

A definição do número de amostras e da forma pela qual o mapeamento deve ser realizado é o passo mais importante para que se atinja sucesso no processo de treinamento da rede neural. Mapeamentos bastante complexos podem ser aproximados por uma rede neural, mas para que se obtenha uma generalização precisa, é necessário que se consiga, de alguma forma, captar a estrutura do sistema não linear pelas conexões da rede neural. Um procedimento bastante simples é usado como diretiva para a definição do número de amostras. Considere um mapeamento não linear $f(x)$, a ser aproximado por uma rede neural. Se $x=x(t)$ for uma função periódica, então $y=f(x(t))$ também será uma função periódica e,

portanto, pode-se obter considerável informação de sua resposta de frequência ou análise espectral. Se a análise for feita usando-se dados amostrados, o índice de Nyquist se constituirá em uma razoável estimativa inicial para a escolha do número de amostras.

Em geral, para sistemas não lineares, tais como sistemas dinâmicos de entrada-saída, descritos por mapeamentos não lineares, não é possível o uso do método de análise de frequência convencional, porque o princípio da superposição não se aplica. Assim sendo, no caso do uso de redes neurais para aproximação funcional de mapeamentos não lineares, a análise espectral deverá somente ser usada para se identificar a frequência harmônica mais significativa e usar esta informação para se determinar o intervalo de amostragem apropriado.

Considere uma sequência de tempo discreto $[y_k, k=0, \dots, m-1]$. Desde que o sinal seja de duração finita, pode-se tomar $2\pi/m$ como a frequência fundamental da sequência (em radianos), para a finalidade de representação funcional. Em geral o inteiro m , pode ser considerado como o período da sequência ou a duração máxima de um sinal considerado não periódico, porém periodicamente estendido em tempo, além das m amostras.

Será mostrado em seguida (veja equação 9) que uma rede neural de várias camadas pode apresentar oscilações de alta frequência quando usada em aproximações funcionais. Embora não haja relação direta entre tais níveis de oscilação e aqueles obtidos pela aproximação através da transformada discreta de Fourier, sugere-se que o índice de Nyquist seja usado como primeira aproximação. Ao final do treinamento da rede, deve-se aplicar um teste de precisão na aproximação funcional. Se o modelo de rede empregado apresentar uma generalização precisa, estará garantido que a mais alta frequência harmônica contida nos dados

estará incluída no modelo.

A sequência $[y_k, k=1, \dots, m]$, pode ser aproximada em termos dos coeficientes de Fourier $A(p)$ por:

$$y(k) = \frac{1}{m} \sum_{p=1}^m A(p) \exp\left(j \frac{2\pi}{m} pk\right) \quad (1)$$

$$A(p) = \sum_{k=1}^m y(k) \exp\left(-j \frac{2\pi}{m} pk\right) \quad (2)$$

$p=1, 2, \dots, m$

Os coeficientes $A(p)$ são números complexos e definem a amplitude complexa do componente harmônico de ordem p do sinal dado $[y_k, k=1, \dots, m]$. Seja $M\tau$ o período fundamental do sinal contínuo $y(t)$, $y_k = y(k\tau)$ e τ o período de amostragem. Desta forma, a mais alta frequência do sinal amostrado, a ser representada pela transformada discreta de Fourier, terá um período igual a τ , que corresponde ao período de amostragem. Considere que uma função contínua esteja sendo aproximada usando-se uma série de senos e cossenos, determinada pela transformada discreta de Fourier. Neste caso, para se evitar *aliasing* no processo de amostragem, a frequência de amostragem deverá ser pelo menos o dobro da maior frequência presente nos dados (índice de Nyquist).

Neste trabalho, está sendo proposto que o mesmo princípio seja aplicado, como ponto de partida, na amostragem de um mapeamento não linear para fins de treinamento de uma rede neural. Como uma aproximação funcional, as redes neurais podem apresentar significativas oscilações dentro do intervalo de amostragem. Além do mais, não há garantia de que estas oscilações poderão se totalmente evitadas usando-se o índice de Nyquist.

As seguintes definições são necessárias para se formalizar os conceitos de rede neural e elementos de uma rede neural.

Definição 1: Um elemento de uma rede neural é um dispositivo de processamento cuja saída y é o resultado da soma ponderada de suas entradas x e uma referência unitária, calculada através de uma função de ativação σ , isto é:

$$y = \sigma \left(\sum_k w_k \cdot x_k + \theta \right) \quad (3)$$

onde $w \in \mathbb{R}^n$ é um vetor contendo o valor das conexões e θ representa um valor de limiar. ■

Definição 2: Uma rede neural é uma estrutura com ou sem camadas intermediárias onde as saídas de cada um dos elementos de uma camada estão conectados a todos os elementos das camadas subsequentes incluindo a saída. Uma classe particular de redes neurais é aquela que possui conexões somente entre camadas adjacentes. ■

Afirmção 1: Embora diversas funções possam ser usadas como função de ativação, a função sigmóide é em geral usada pelo fato de ser diferenciável e monótona. Estas características são importantes para satisfazer as condições de treinabilidade da rede neural (veja teorema 1). ■

Considere inicialmente uma rede neural com somente uma camada intermediária de n elementos. Seja $\tilde{y}(x)$ a saída da rede treinada com os padrões de entrada-saída $\{(k\tau, y(k)), k=1, \dots, m\}$. Suponha que f seja uma função contínua, amostrada uniformemente com período τ , isto é: $f(k\tau) = y(k), k=1, \dots, m$.

Se n for o número de elementos na camada intermediária, a saída da rede será:

$$\tilde{y}(x) = \sigma\left(\sum_{k=1}^n Q_k \sigma(P_k x + \alpha_k) + \beta\right) \quad (4)$$

onde σ é a função de ativação (6), P_k , α_k , Q_k , β são os pesos e valores de limiares que conectam a entrada à camada intermediária e a camada intermediária à camada de saída, respectivamente. A saída da rede neural, para um ponto situado dentro do intervalo de amostragem (interpolação) pode ser calculada por:

$$\tilde{y}(x+\Delta x) = \sigma\left(\sum_{k=1}^n Q_k \sigma(P_k (x+\Delta x) + \alpha_k) + \beta\right) \quad (5)$$

onde, $\sigma(\lambda) = 1/(1+\exp(-\lambda))$ (6)

e Δx é a distância do ponto em teste ao ponto amostrado mais próximo.

O somatório na equação (4) representa a entrada líquida do elemento de saída da rede neural, isto é:

$$\text{net}(x) = \sum_{k=1}^n Q_k \frac{1}{1 + \exp(-(P_k x + \alpha_k))} \quad (7)$$

A derivada da equação (7) pode ser determinada por:

$$d \text{net}(x)/dx = \text{net}(x)[1-\text{net}(x)] \quad (8)$$

Verifica-se, portanto, que o valor máximo da derivada dada em (8), $(\frac{d \text{net}(x)}{dx} \text{ max})$ depende dos valores dos pesos. Assim sendo, a amplitude máxima de oscilação entre dois pontos amostrados estará limitada por:

$$\tilde{y}(x+\Delta x)_{MAX} = \sigma\left(\left(\frac{d \text{net}(x)}{dx} \max\right)\Delta x + \beta\right) \quad (9)$$

A equação (9) mostra que, dentro do intervalo de amostragem, a amplitude da aproximação pode ser inaceitável dependendo dos pesos das conexões. Verifica-se também que, se o número de elementos da camada intermediária for aumentado, haverá, como consequência, uma maior tendência para oscilações indesejáveis. Apesar das amplitudes de tais oscilações serem imprevisíveis, o índice de Nyquist representa uma diretiva bastante útil para se determinar o número de padrões de treinamento. Depois do processo de aprendizagem, deve-se testar a rede neural em pelo menos um ponto dentro de cada intervalo de amostragem. Se o erro de saída não corresponder a tolerância especificada, então o número de amostras deverá ser aumentado. Observa-se contudo que, quando se aumenta o número de amostras, há necessidade de se aumentar o número de elementos da camada intermediária.

A análise esboçada acima pode ser estendida para redes ou sistemas de múltiplas entradas usando-se o Teorema de Kolmogorov, veja Kolmogorov (1957), o qual estabelece: "Para $I=[0,1]$, dada uma função contínua f em I^N , esta pode ser escrita na forma:

$$f(x_1, \dots, x_N) = \sum_{q=1}^{2N+1} g_q \left(\sum_{p=1}^N \phi_{pq}(x_p) \right) \quad (10)$$

onde g_p e ϕ_{pq} são funções contínuas apropriadamente escolhidas de uma única variável e ϕ_{pq} são funções crescentes".

Isto significa que uma função de múltiplas entradas pode ser transformada em uma combinação de funções de entradas únicas, sob condições apropriadas, pela escolha correta de g_q e ϕ_{pq} .

A próxima seção apresenta diretivas para se determinar a estrutura apropriada da rede neural.

2. Definição da Estrutura de Redes Neurais de Múltiplas Camadas

A base teórica para o uso de redes neurais de múltiplas camadas para aproximação funcional é dada em Hornik (1989), teorema 2.5, onde se prova que redes neurais de múltiplas camadas são aproximadores universais. Entretanto, a determinação da arquitetura da rede neural deve incluir considerações de treinabilidade. Nas proposições seguintes são apresentadas condições para se determinar o número de elementos em cada camada intermediária, embora, muitas vezes sejam necessários ajustes posteriores pois, para mapeamentos não lineares, não há garantia de que o número de elementos seja suficiente para representar a estrutura do sistema.

Teorema 1: Dados m conjuntos (padrões de treinamento) consistindo de múltiplas entradas e saída única, um número suficiente de parâmetros de ajuste em uma rede neural com uma camada intermediária, para representar exatamente a relação de entrada-saída (para as amostras apresentadas) é igual a m , considerando-se que a função de ativação seja da classe C' em \mathbb{R}^n (uma função é da classe C' em \mathbb{R}^n se as primeiras derivadas parciais existirem e forem contínuas).

Demonstração: considere uma rede neural com n_1 entradas, uma saída e n elementos em uma única camada intermediária. O conjunto de equações de saída para m amostras é o seguinte:

$$\begin{aligned}
y(1) - \sigma(Q_1 \sigma(P_{1,1} \cdot x_1(1) + \dots + P_{1,n_I} \cdot x_{n_I}(1) + \alpha_1) + \dots \\
+ Q_n \sigma(P_{n,1} \cdot x_1(1) + \dots + P_{n_I,n} \cdot x_{n_I}(1) + \alpha_n) + \beta) = 0 \\
\dots \qquad \qquad \qquad (11)
\end{aligned}$$

$$\begin{aligned}
y(m) - \sigma(Q_1 \sigma(P_{1,1} \cdot x_1(m) + \dots + P_{1,n_I} \cdot x_{n_I}(m) + \alpha_1) + \dots \\
+ Q_n \sigma(P_{n,1} \cdot x_1(m) + \dots + P_{n_I,n} \cdot x_{n_I}(m) + \alpha_n) + \beta) = 0
\end{aligned}$$

onde $P_{1,j}$ é o peso da conexão entre a entrada j e o elemento 1 da camada intermediária; Q_k é o peso da conexão entre o elemento k da camada intermediária e a saída; α_k é o valor de limiar do elemento k ; β é o limiar de saída.

Para simplificar a notação, seja $w \in \mathbb{R}^T$ um vetor representando todos os pesos na rede neural, isto é:

$$w = (Q_1, P_{1,1}, \dots, P_{1,n_I}, \alpha_1, \dots, Q_n, P_{n,1}, \dots, P_{n_I,n}, \alpha_n, \beta) \quad (12)$$

e considere que o conjunto (11) seja representado pelas funções não lineares ϕ_1, \dots, ϕ_m :

$$\begin{aligned}
\phi_1(w_1, \dots, w_T) = 0 \\
\dots \\
\phi_m(w_1, \dots, w_T) = 0
\end{aligned} \qquad \qquad \qquad (13)$$

Basicamente, o processo de treinamento da rede neural consiste em se resolver o conjunto (13) em w . Além do mais, pelo teorema da função implícita (Buck, 1978, pag 362), dado o conjunto

(13) em T variáveis, e um ponto $\bar{w} = (\bar{w}_1, \dots, \bar{w}_T)$ que satisfaça o sistema, pode-se (teóricamente) resolver o sistema (13), para um conjunto de variáveis, w_1, \dots, w_m , em função das variáveis restantes, nas proximidades de \bar{w} , se o determinante do Jacobiano,

$$\det J = \left| \frac{\partial(\phi_1, \phi_2, \dots, \phi_m)}{\partial(w_1, \dots, w_T)} \right| \neq 0 \quad (14)$$

em \bar{w} . Consequentemente, a condição necessária para a unicidade da solução é que o número T de parâmetros de ajuste seja igual a m . Para o caso $T > m$, haverá infinitas soluções possíveis. Como as funções de ativação são monótonas e de classe C' , isto garante que as derivadas das funções de ativação nunca serão anuladas. Portanto, mesmo que (14) não seja satisfeita, existirá sempre uma solução, não necessariamente única. ■

Para as redes de múltiplas saídas, o conjunto de equações será aumentado por um fator igual ao número de saídas. O seguinte corolário formaliza a condição de existência para o caso de múltiplas saídas.

Corolário 1: Dados m padrões de treinamento com múltiplas entradas e múltiplas saídas (n_o saídas), um número suficiente de parâmetros de ajuste em uma rede neural com uma única camada intermediária, para representar exatamente a relação de entrada-saída (para o conjunto de amostras apresentado) é igual a $n_o m$, desde que as funções de ativação sejam monótonas e de classe C' . ■

Do Teorema 1 e Corolário 1, pode-se concluir que a existência de solução não depende do número de padrões de treinamento mas do número de parâmetros de ajuste. Portanto, pode-se definir um número suficiente de elementos na camada

onde as matrizes A e B estarão na forma canônica de Brunovsky. Isidori (1989) apresenta os detalhes de obtenção da mudança de coordenadas e da função de linearização. Na próxima seção, serão analisados sistemas de tempo discreto do tipo SISO.

3. Linearização por realimentação de sistemas de tempo discreto do tipo SISO

Nesta seção, inicialmente são analisadas as condições para linearização por realimentação de um sistema de tempo discreto não linear. Em seguida, são revistos o método para determinação do grau relativo e os procedimentos para a determinação da função de realimentação e da mudança de coordenadas, capazes de linearizar este tipo de sistema.

A crescente aplicação de dispositivos de controle de tempo discreto em instalações industriais tem intensificado o interesse no projeto de controladores digitais não lineares. A maioria dos trabalhos anteriores se dedicaram a sistemas lineares. O assunto tem sido investigado teoricamente por vários autores, porém apenas alguns resultados práticos foram atingidos para classes gerais de sistemas não lineares. Sendo assim, tem havido um grande interesse na extensão do método de linearização por realimentação, para sistemas discretos. Para o caso de sistemas amostrados, que constituem a maioria dos sistemas de tempo discreto, Jakubczyk (1987) mostrou que linearização através de realimentação digital impõe restrições altamente não genéricas na estrutura do sistema. Estas restrições irão ocorrer mesmo que o sistema contínuo original seja linearizável por realimentação. Grizzle (1988) estabeleceu um conjunto de condições necessárias e suficientes para linearização local por realimentação em uma classes específica de sistemas de tempo discreto não linear.

Nesta seção, inicia-se com uma discussão das condições estabelecidas por Grizzle e Jakubczyk, e são apresentados os procedimentos para linearização de sistemas de tempo discreto SISO. No capítulo IV mostra-se que, satisfeitas algumas condições, a função de controle que lineariza o sistema, por realimentação, pode ser estimada por uma rede neural de múltiplas camadas.

3.1 Definições

Dois tipos de sistemas não lineares de tempo discreto, descritos pelas equações (12) e (16), onde $m=p=1$, são analisados nesta seção. Para o sistema Σ_N , considera-se que X , U e Y sejam *manifolds* suaves (veja apêndice II) de dimensões finitas n , m e p respectivamente. As funções ζ e h são mapeamentos suaves em:

$$\zeta : X \times U \rightarrow X \quad (71)$$

$$h : X \rightarrow Y \quad (72)$$

Analisa-se também nesta seção, uma classe específica constituída de sistemas não lineares de tempo discreto, onde a entrada aparece de forma linear e que podem ser descritos pela equação (12), onde $m=p=1$, no caso de sistemas SISO.

Um dos procedimentos para linearizar um sistema de tempo discreto, consiste em se definir a seguinte função, passível de inverão:

$$\lambda : X \times U \rightarrow U \quad (73)$$

que mapea a entrada e o estado no sinal atuante, tal que:

$$x(k+1) = \zeta(x(k), \lambda(x(k), v(k))) \quad (74)$$

$$y(k) = h(x(k)) \quad (75)$$

e

$$v(k) = \lambda^{-1}(x(k), u(k)) \quad (76)$$

Desta forma, o sistema não linear original, com o mesmo desempenho da resposta entrada-estado-saída, pode se representado como na figura 5.

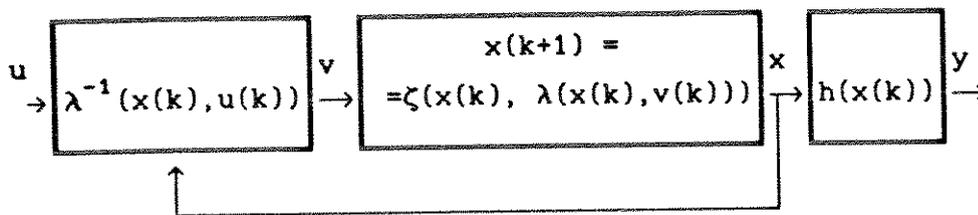


Fig. 5 - Aplicação da função inversa como controlador para o sistema não linear.

Considere que seja aplicada uma mudança de coordenadas no espaço de estado para uma outra variável z , através de um mapeamento passível de inversão, tal que:

$$z(k) = \phi(x(k)) \quad (77)$$

A aplicação de uma mudança de coordenadas transforma o diagrama anterior em:

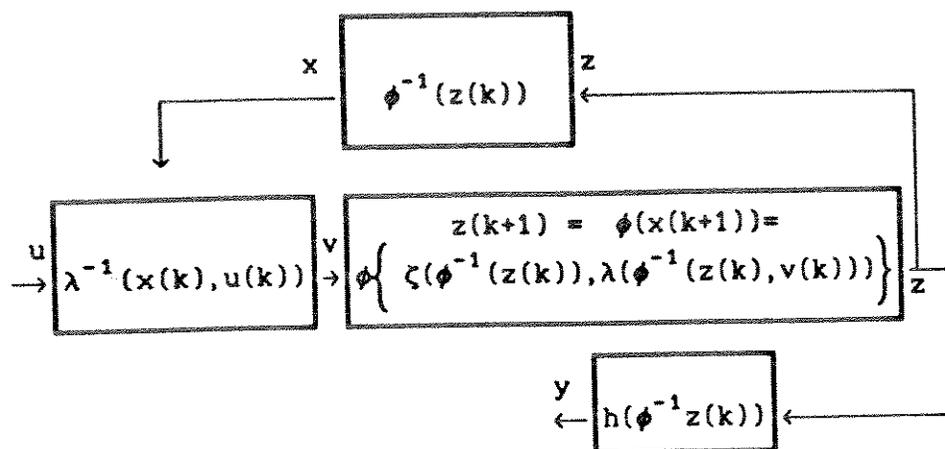


Fig 6 - Aplicação de uma mudança de coordenadas no sistema não linear.

Considere que ϕ e λ sejam definidas de tal forma que:

$$\phi \left\{ \zeta(\phi^{-1}(z(k)), \lambda(\phi^{-1}(z(k), v(k)))) \right\} = A z(k) + B v(k) \quad (78a)$$

$$h_D(\phi^{-1}(z(k))) = c^T z(k) \quad (78b)$$

Neste caso, pode-se dizer que o sistema foi linearizado em termos da resposta (v-z-y) entrada-estado-saída, considerando-se satisfeitas certas condições, descritas a seguir.

O diagrama anterior, embora matematicamente correto, é geralmente impraticável de se aplicar a uma instalação industrial devido a dificuldade de se separar a dinâmica do sistema da função de entrada e saída. Considerando-se contudo, que o estado do sistema seja observável, o diagrama da figura 7 mostra como se implementar a mudança de coordenadas e a realimentação de estado.

intermediária baseado no Teorema 1. Os Corolários 2 e 3 especificam o número de elementos em redes de uma ou duas camadas intermediárias .

Corolário 2: Dado um conjunto de m padrões de treinamento, o número suficiente n de elementos na camada intermediária em uma rede neural com uma única camada intermediária, capaz de representar a relação de entrada-saída é:

$$n = \text{int}\left\{ \frac{(m-1)n_o}{n_I + n_o + 1} \right\} \quad (15)$$

onde n_I é o número de elementos de entrada, n_o é o número de saídas das rede, e int representa o próximo inteiro. ■

Corolário 3: Dado um conjunto de m padrões de entrada-saída, o número suficiente de elementos nas camada intermediárias em uma rede neural com duas camadas intermediárias, para representar exatamente a relação entrada-saída, deve satisfazer:

$$n_o m \leq n_o(n_2 + 1) + n_1(n_I + 1) + n_2(n_1 + 1) \quad (16)$$

onde n_I é a dimensão de entrada, n_o é a dimensão saída, n_1 e n_2 são os números de elementos da primeira e da segunda camadas, respectivamente. ■

Estes resultados podem ser expressos através dos seguintes procedimentos.

3. Procedimentos para Identificação de Mapeamentos Não-Lineares Usando-se Redes Neurais

Os resultados obtidos das investigações apresentadas

acima, podem ser resumidos nos seguintes procedimentos, a serem usados na solução do problema de se amostrar um dado sinal, para identificação de mapeamentos não lineares, através de redes neurais.

1. Identifique a mais alta frequência harmônica envolvida;
2. Use o índice de Nyquist para determinar o taxa de amostragem do referido sinal;
3. Estime o número de elementos da camada intermediária ;
4. Treine a rede neural;
5. Teste a precisão de generalização em pontos dentro do período de amostragem.
6. Se a tolerância desejada não for atingida, diminua o período de amostragem e volte a passo 1, caso contrário, pare.

No tipo de treinamento apresentado, um certo número de padrões são apresentados à rede, repeditamente até que a precisão desejada seja atingida.

Os procedimentos aqui apresentados e ilustrados em Teixeira(1991) se referem ao caso em que o treinamento das redes é realizado em um esquema off-line, em relação ao sistema físico em questão. Nesse tipo de treinamento, um certo número de padrões são apresentados à rede, repeditamente até que a precisão desejada seja atingida. O treinamento on-line das redes neurais é comentado a seguir.

4. Treinamento On-line de Redes Neurais

Considere que a rede neural esteja conectada diretamente ao sistema físico através de uma interface de aquisição de dados. Os valores são colhidos e, a cada intervalo de amostragem, são apresentados à rede, onde um passo de propagação retroativa é

realizado. Considerando-se não haver periodicidade na excitação do sistema físico, não haverá também periodicidade no treinamento da rede que receberá, a cada apresentação valores diferentes para treinamento. Assim sendo, os procedimentos apresentados anteriormente não se aplicam. Entretanto, se o número de elementos da camada intermediária for suficiente, pode-se atingir valores de erro de treinamento que satisfaçam a precisão requerida.

Por outro lado, deve-se considerar que a aproximação funcional por redes neurais é equivalente a uma aproximação por uma série limitada. Desta forma, aumentando-se o número de elementos das camadas intermediárias, em geral melhora-se a aproximação da função. Portanto, para o caso de treinamento on-line, deve-se estabelecer uma solução de compromisso entre a precisão desejada e o tempo disponível para treinamento que, por sua vez, está limitado pelo intervalo de amostragem. Se a precisão desejada não for atingida, mesmo após várias apresentações, deve-se aumentar o número de elementos das camadas intermediárias. Deve-se considerar também que, em casos especiais, o aumento do número de elementos das camadas intermediárias não implica em melhoria de precisão.

As simulações apresentadas nos capítulos III e IV ilustram o treinamento das redes neurais aplicadas na identificação de sistemas dinâmicos não lineares.

5. Conclusões

Neste capítulo, foi apresentado um método para se avaliar do número de amostras e o número de unidades das camadas intermediárias em uma arquitetura de rede neural, para aproximação funcional. Foram também apresentadas considerações com relação ao treinamento on-line de redes neurais conectadas a sistemas dinâmicos, estabelecendo-se assim uma estrutura matemática para as simulações apresentadas nos capítulos III e IV.

CAPITULO II

Linearização por Realimentação em Sistemas Dinâmicos não Lineares

Nas últimas décadas, a teoria de sistemas de controle lineares tem sido objeto de intensa pesquisa e tem influenciado decisivamente a maioria das aplicações industriais. Embora os sistemas físicos sejam intrinsecamente não lineares, técnicas lineares são usadas através de métodos de linearização.

Recentemente, tem havido um grande interesse na análise de sistemas não lineares e na síntese e aplicação de estratégias de controle que sejam inerentemente não lineares. Uma das técnicas em desenvolvimento, é a linearização exata por realimentação, na qual o sistema não linear é transformado em um sistema linear através de uma mudança de coordenadas e por realimentação de estado não linear. Este processo tem a vantagem de que, após a linearização, o sistema possa ser controlado usando-se métodos lineares de controle.

Um dos objetivos deste capítulo é apresentar uma análise dos vários procedimentos envolvendo linearização por realimentação, além de adicionar considerações relacionadas ao objetivo principal deste trabalho, isto é, linearização por realimentação usando-se redes neurais.

Inicialmente, são apresentados conceitos básicos tais como, grau relativo e dinâmica nula para sistemas não lineares de tempo discreto e contínuo, SISO e MIMO. Estes conceitos formam a base teórica das técnicas de linearização por realimentação, e portanto, torna-se necessário uma perfeita compreensão destes princípios para se determinar as limitações do método e os detalhes de implementação.

Os seguintes tipos de sistema são considerados neste trabalho:

Sistema de tempo contínuo ψ :

Considere um sistema com m entradas, $u_1(t), \dots, u_m(t)$, e p saídas, $y_1(t), \dots, y_p(t)$, onde t é um parâmetro contínuo representando a variável independente tempo. Para simplificar a notação, a variável t é omitida neste trabalho. O sistema é descrito por:

$$\psi: \begin{cases} \dot{x} = f(x) + \sum_{i=1}^m g_i(x)u_i & (01a) \end{cases}$$

$$\begin{cases} y = h(x) & (01b) \end{cases}$$

$$\text{O vetor de estado, } x = (x_1, \dots, x_n) \quad (02)$$

pertence a um conjunto aberto X de \mathbb{R}^n e f, g_1, \dots, g_m são mapeamentos em \mathbb{R}^n definidos no conjunto aberto X . $f(x)$ representa o valor de f no ponto x de X . A mesma consideração se faz para $g_1(x), \dots, g_m(x)$, isto é, $g_1(x)$ representa o valor de g_1 no ponto x de X . O mapeamento de saída h é uma função em \mathbb{R}^p , também definida no conjunto aberto X de \mathbb{R}^n . $h(x)$ representa o valor de h no ponto x de X . As funções f, g_1, \dots, g_m , correspondem a mapeamentos de \mathbb{R}^n para \mathbb{R}^n e podem ser escritas em forma de coordenadas, como funções das variáveis x_1, \dots, x_n , isto é:

$$f(x) = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ f_2(x_1, \dots, x_n) \\ \dots \\ f_n(x_1, \dots, x_n) \end{bmatrix} \quad (03)$$

$$g_1(x) = \begin{bmatrix} g_{11}(x_1, \dots, x_n) \\ g_{12}(x_1, \dots, x_n) \\ \dots \\ g_{1n}(x_1, \dots, x_n) \end{bmatrix} \quad (04)$$

$$g_2(x) = \begin{bmatrix} g_{21}(x_1, \dots, x_n) \\ g_{22}(x_1, \dots, x_n) \\ \dots \\ g_{2n}(x_1, \dots, x_n) \\ \dots \end{bmatrix} \quad (05)$$

$$g_m(x) = \begin{bmatrix} g_{m1}(x_1, \dots, x_n) \\ g_{m2}(x_1, \dots, x_n) \\ \dots \\ g_{mn}(x_1, \dots, x_n) \end{bmatrix} \quad (06)$$

O mapeamento h pode também ser representado na forma de um vetor de dimensão p das variáveis x_1, \dots, x_n , isto é:

$$h(x) = \begin{bmatrix} h_1(x_1, \dots, x_n) \\ h_2(x_1, \dots, x_n) \\ \dots \\ h_p(x_1, \dots, x_n) \end{bmatrix} \quad (07)$$

Supõe-se que os mapeamentos f, g_1, \dots, g_m e h sejam funções suaves de seus argumentos (veja apêndice II). Isto significa que $f_1, \dots, f_n, g_{11}, \dots, g_{mn}, h_1, \dots, h_p$ possuem derivadas parciais contínuas de qualquer ordem. Supõe-se também, quando necessário, que as funções sejam analíticas (veja apêndice II). Neste caso o sistema é denominado Sistema Analítico Linear. O termo linear se refere ao fato de que os controles u_1, \dots, u_m entram linearmente no sistema.

Sistema de Tempo Contínuo ψ_N :

Este sistema é mais geral e os controles não entram linearmente no modelo:

$$\psi_N: \left\{ \begin{array}{l} \dot{x} = \zeta(x, u) \\ y = h(x) \end{array} \right. \quad \begin{array}{l} (08a) \\ (08b) \end{array}$$

O estado $x = (x_1, \dots, x_n)$, também pertence a um conjunto aberto X de \mathbb{R}^n e a entrada $u = (u_1, \dots, u_m)$ pertence a um conjunto aberto U de \mathbb{R}^m . A função ζ é um mapeamento em \mathbb{R}^n do produto espacial $\mathbb{R}^n \times \mathbb{R}^m$ e o mapeamento de saída h é uma função em \mathbb{R}^p também definida no conjunto aberto X de \mathbb{R}^n .

Sistema Contínuo de entrada-saída ψ_y :

Eventualmente, são considerados sistemas descritos por equações diferenciais de entrada e saída, isto é:

$$\text{Sistema } \psi_y: y^{(n)} = \theta(y, y^{(1)}, \dots, y^{(n-1)}, u) \quad (09)$$

onde a saída y e suas n derivadas $y^{(1)}, y^{(2)}, \dots, y^{(n)}$ pertencem a um conjunto aberto Y de \mathbb{R}^p e a entrada u pertence a um conjunto aberto U de \mathbb{R}^m .

Sistemas lineares também são considerados neste trabalho e podem ser descritos pela seguinte equação de estado:

$$\text{Sistema linear } \mathcal{L}: \begin{cases} \dot{z} = Az + Bv & (10a) \\ y = Cz & (10b) \end{cases}$$

onde $v \in \mathbb{R}^m$ é a entrada, $y \in \mathbb{R}^p$ é a saída. A , B e C são matrizes constantes de dimensões $n \times n$, $n \times m$ e $p \times n$ respectivamente e $z \in \mathbb{R}^n$ é a variável de estado.

Sistemas lineares também podem ser descritos por equações diferenciais de entrada e saída, tais como:

Sistema linear \mathcal{L}_y :

$$y^{(n)} = A_1 y^{(0)} + A_2 y^{(1)} + \dots + A_n y^{(n-1)} + Bu \quad (11)$$

onde $y \in \mathbb{R}^p$ é a saída, $y^{(1)}, y^{(2)}, \dots, y^{(n)} \in \mathbb{R}^p$ são derivadas da saída, $u \in \mathbb{R}^m$ é a entrada. A_1, \dots, A_n são matrizes constantes $p \times p$ e B é uma matriz constante $p \times m$.

Sistema de tempo discreto Σ :

Considerando-se k o instante de tempo e δ o intervalo entre dois instantes sucessivos, tem-se $t=k\delta$, e o sistema poderá então ser descrito por:

$$\text{Sistema } \Sigma: \begin{cases} x(k+1) = f(x(k)) + \sum_{i=1}^m g_i(x(k))u_i(k) & (12a) \\ y(k) = h(x(k)) & (12b) \end{cases}$$

O estado $x=(x_1, \dots, x_n)$ pertence a um conjunto aberto X de \mathbb{R}^n e f, g_1, \dots, g_m são mapeamentos em \mathbb{R}^n definidos em um conjunto aberto X . Por $f(x(k))$ representa-se o valor de f no ponto x de X no instante k . Da mesma forma, $g_1(x(k)), \dots, g_m(x(k))$ representam o valor de g_i no ponto x de X , no instante k . O mapeamento de saída h é uma função em \mathbb{R}^p também definida em um conjunto aberto X de \mathbb{R}^n e $h(x(k))$ representa o valor de h no ponto x de X no instante k . Os mapeamentos f, g_1, \dots, g_m , também podem ser representados na seguinte forma vetorial:

$$f(x(k)) = \begin{bmatrix} f_1(x_1(k), \dots, x_n(k)) \\ f_2(x_1(k), \dots, x_n(k)) \\ \dots \\ f_n(x_1(k), \dots, x_n(k)) \end{bmatrix} \quad (13)$$

$$g_1(x(k)) = \begin{pmatrix} g_{11}(x_1(k), \dots, x_n(k)) \\ g_{12}(x_1(k), \dots, x_n(k)) \\ \dots \\ g_{1n}(x_1(k), \dots, x_n(k)) \end{pmatrix} \quad (14a)$$

$$g_2(x(k)) = \begin{pmatrix} g_{21}(x_1(k), \dots, x_n(k)) \\ g_{22}(x_1(k), \dots, x_n(k)) \\ \dots \\ g_{2n}(x_1(k), \dots, x_n(k)) \end{pmatrix} \quad (14b)$$

$$\dots$$

$$g_m(x(k)) = \begin{pmatrix} g_{m1}(x_1(k), \dots, x_n(k)) \\ g_{m2}(x_1(k), \dots, x_n(k)) \\ \dots \\ g_{mn}(x_1(k), \dots, x_n(k)) \end{pmatrix} \quad (14c)$$

O mapeamento h pode ser representado na forma de um vetor p -dimensional das variáveis reais x_1, \dots, x_n , em cada instante k , isto é:

$$h(x(k)) = \begin{pmatrix} h_1(x_1(k), \dots, x_n(k)) \\ h_2(x_1(k), \dots, x_n(k)) \\ \dots \\ h_p(x_1(k), \dots, x_n(k)) \end{pmatrix} \quad (15)$$

Supõe-se que os mapeamentos f, g_1, \dots, g_m e h sejam funções suaves em seus argumentos (veja apêndice II), ou seja,

$f_1, \dots, f_n, g_{11}, \dots, g_{mn}, h_1, \dots, h_p$ sejam funções reais do estado x , possuindo derivadas parciais contínuas de qualquer ordem. Sempre que necessário, as funções f, g_1, \dots, g_m , são consideradas analíticas (veja apêndice II), e neste caso, o sistema Σ é considerado Sistema de Tempo Discreto Linear Analítico.

Sistema de tempo discreto Σ_N :

Esta é uma classe mais geral de sistemas de tempo discreto, representada por um conjunto de equações à diferença não lineares, onde o controle não aparece linearmente.

$$\Sigma_N: \begin{cases} x(k+1) = \zeta(x(k), u(k)) & (16a) \\ y(k) = h(x(k)) & (16b) \end{cases}$$

A variável de estado $x(k) = (x_1(k), \dots, x_n(k))$, pertence a um conjunto aberto X de \mathbb{R}^n e a entrada $u(k) = (u_1(k), \dots, u_m(k))$ pertence a um conjunto aberto U de \mathbb{R}^m . A função ζ corresponde a um mapeamento em \mathbb{R}^n definido em $\mathbb{R}^n \times \mathbb{R}^m$ e a função de saída h corresponde a um mapeamento em \mathbb{R}^p definido em um conjunto aberto X de \mathbb{R}^n .

Um sistema não linear de tempo discreto pode também ser descrito por um sistema de equações de entrada e saída, isto é:

Sistema Σ_y : $y(k+1) = \eta(y(k), y(k-1), \dots, y(k-n+1), u(k))$ (17)

onde $y(k+1), y(k), y(k-1), \dots, y(k-n+1)$ são valores de saída medidos nos instantes $k+1, k, k-1, \dots, k-n+1$, respectivamente, pertencendo ao

conjunto aberto Y de \mathbb{R}^p . A entrada u pertence a um conjunto aberto U de \mathbb{R}^m .

Sistemas lineares de tempo discreto, descritos pela seguinte equação de estado, são também considerados neste trabalho:

$$\text{Sistema Linear } \Gamma: \begin{cases} z(k+1) = Az(k) + Bv(k) & (18a) \\ y(k) = Cz(k) & (18b) \end{cases}$$

onde $z \in \mathbb{R}^n$, $y \in \mathbb{R}^p$, $v \in \mathbb{R}^m$ representam as variáveis de estado, de saída e de entrada, respectivamente. A , B , e C são matrizes constantes, de dimensões $n \times n$, $n \times m$ e $p \times n$, respectivamente.

Eventualmente, um sistema linear pode ser representado pela seguinte equação de entrada e saída:

$$\text{Sistema Linear } \Gamma_y: \\ y(k+1) = A_0 y(k) + A_1 y(k-1) + \dots + A_{n-1} y(k-n+1) + Bv(k) \quad (19)$$

onde $y \in \mathbb{R}^p$ e $v \in \mathbb{R}^m$ são os vetores de entrada e saída, respectivamente. A_0, A_1, \dots, A_{n-1} são matrizes constantes de dimensão $p \times p$ e B é uma matriz constante de dimensão $p \times m$.

Na próxima seção, será analisada a linearização por realimentação de sistemas contínuos de entrada e saída únicas (SISO).

1. Linearização por Realimentação em Sistemas SISO de Tempo Contínuo Não-Lineares

Nesta seção é apresentada uma revisão das condições e conceitos básicos de linearização por realimentação em sistemas de tempo contínuo (Isidori, 1989).

Basicamente, são analisados sistemas tipo ψ , equação (1), com a restrição $m=p=1$. Assim sendo, nesta seção, ψ representa um sistema SISO, isto é:

$$\psi: \begin{cases} \dot{x}(t) = f(x(t)) + g(x(t))u(t) & (20a) \\ y(t) = h(x(t)) & (20b) \end{cases}$$

Em geral, a variável independente t é omitida e o sistema fica:

$$\psi: \begin{cases} \dot{x} = f(x) + g(x)u & (21a) \\ y = h(x) & (21b) \end{cases}$$

Sastry (1989) apresenta uma breve revisão do assunto e considera que o primeiro trabalho em linearização de entrada-saída por realimentação se deve a Porter (1970). Subseqüentes trabalhos foram publicados por Singh (1975), Freund (1975) e Isidori (1981). Uma das primeiras implementações práticas do método de linearização por realimentação consistiu no controle de dinâmica de voo, mencionado por Meyer (1980).

Duas técnicas são consideradas nesta seção:

- (a). Linearização exata de entrada-saída por realimentação.
- (b). Linearização exata por realimentação de estado.

Na técnica (a), através de realimentação não linear, o sistema se torna linear sob o ponto de vista de entrada-saída. A técnica (b) por outro lado, necessita de uma mudança de coordenadas no espaço de estado e uma realimentação de estado não linear. Desta forma, o sistema ψ é transformado em um sistema linear em termos das respostas de entrada-saída e entrada-estado.

Os primeiros desenvolvimentos nesta área se concentraram na técnica (a), até o aparecimento dos trabalhos desenvolvidos independentemente por Jakubczyk (1987) e Hunt (1983), onde o problema de linearização exata, por realimentação no espaço de estado, foi solucionado para uma classe restrita de sistemas. Para se discutir esta técnica, o conceito de grau relativo em sistemas contínuos será inicialmente definido.

1.1 Conceito de Grau-Relativo

O grau relativo do sistema ψ corresponde ao número de vezes que a função de saída tem que ser derivada em relação ao tempo, para que a entrada apareça explicitamente. Derivando-se y em (1), obtém-se:

$$\dot{y} = \frac{\partial h}{\partial x} \cdot \frac{dx}{dt} = \frac{\partial h}{\partial x} (f(x) + g(x)u) =$$
$$\stackrel{\Delta}{=} L_f h(x) + L_g h(x) u \quad (22)$$

onde $L_f h$ e $L_g h$ representam a derivada Lie do campo vetorial h com

relação aos campos vetoriais f e g , respectivamente (veja Apêndice II). Se o grau relativo for maior que 1, então:

$$L_g h(x) = 0, \quad (23)$$

nas proximidades do ponto,

$$x^0 = (x_1(0), x_2(0), \dots, x_n(0))^T \quad (24)$$

e então: $\dot{y} = L_f h(x) \quad (25)$

Para se verificar se o grau relativo é maior que 2, determina-se a segunda derivada, isto é:

$$\begin{aligned} \ddot{y} &= \frac{d^2 y(t)}{dt^2} = \frac{dL_f h(x)}{dt} = \\ &= \frac{\partial L_f h(x)}{\partial x} \frac{dx}{dt} = \frac{\partial L_f h(x)}{\partial x} (f(x) + g(x)u) \end{aligned} \quad (26)$$

ou, $\frac{dL_f h(x)}{dt} = L_f^2 h(x) + L_g L_f h(x)u \quad (27)$

Se o grau relativo for maior que 2, então nas proximidades do ponto x^0 , tem-se $L_g L_f h(x) = 0$, e portanto:

$$\ddot{y} = L_f^2 h(x) \quad (28)$$

Continuando-se desta forma, se r for o primeiro inteiro tal que $L_g L_f^{r-1} h(x^0) \neq 0$, então se diz que o grau relativo do sistema é igual a r , ou seja, o sistema ψ tem grau relativo r no ponto x^0 se:

$$(1) \quad L_g L_f^k h(x^0) = 0, \quad k < r-1 \quad (29)$$

$$(2) \quad L_g L_f^{r-1} h(x^0) \neq 0 \quad (30)$$

Na próxima seção discute-se, como o sistema ψ pode ser linearizado em termos do modelo de entrada-saída; em seguida a linearização no espaço de estado é apresentada.

1.2 Linearização por realimentação do modelo de entrada-saída

Se um sistema tipo ψ SISO tiver grau relativo igual a 1, então $L_g h(x^0) \neq 0$, e a derivada da saída em (1) torna-se:

$$\dot{y} = L_f h(x^0) + L_g h(x^0) u \quad (31)$$

Seja $v(t)$ um sinal de entrada exógeno e considere que o entrada $u(t)$ seja determinada por:

$$u = \frac{1}{L_g h(x)} (-L_f h(x) + v) \quad (32)$$

para qualquer x nas proximidades de x^0 .

Desta forma, da entrada v para a saída y , o desempenho local do sistema pode ser descrito pela seguinte equação diferencial linear:

$$\dot{y} = v \quad (33)$$

Para o caso de grau relativo igual a 2, isto é, $L_g h(x) = 0$, pode-se diferenciar a saída mais uma vez, obtendo-se:

$$\dot{y} = L_f^2 h(x) + L_g L_f h(x) u \quad (34)$$

Novamente, se a entrada for:

$$u = \frac{1}{L_g L_f h(x)} (-L_f^2 h(x) + v) \quad (35)$$

então, da entrada v para a saída y , o desempenho local do sistema poderá ser descrito por:

$$\ddot{y} = v \quad (36)$$

Para o caso geral, onde o grau relativo é igual a r , e portanto, $L_g L_f^{r-1} h(x) \neq 0$, a lei de controle:

$$u = \frac{1}{L_g L_f^{r-1} h(x)} (-L_f^r h(x) + v) \quad (37)$$

transforma o sistema ψ , do ponto de vista de entrada-saída, no seguinte sistema linear:

$$y^{(r)} = v, \text{ onde, } y^{(r)} = \frac{d^r y(t)}{dt^r} \quad (38)$$

O diagrama na figura 1 ilustra a aplicação da realimentação de estado (eq. 37), que transforma o sistema no modelo linear de entrada-saída descrito por (36).

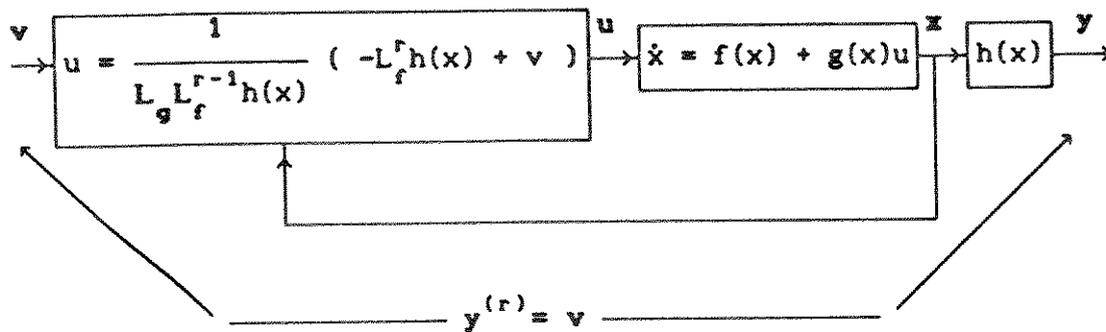


Fig. 1 - Linearização exata de entrada-saída por realimentação de estado

Na próxima seção, é revisto o conceito de linearização exata por realimentação, no espaço de estado.

1.3 Linearização exata por realimentação no espaço de estado

Linearização exata no espaço de estado consiste na aplicação de uma realimentação de estado não linear e uma mudança de coordenadas, para transformar o sistema dado em um sistema linear equivalente em relação a resposta entrada-estado e entrada-saída. Novamente, o conceito de grau relativo se torna relevante para a aplicação deste método. São analisados dois casos onde se considera que o grau relativo é menor ou igual a ordem do sistema. Sem perda de generalização, será considerado que o sistema é de realização mínima (veja seção 1.4).

Caso $r=n$: Para se obter a mudança de coordenadas, calcula-se a derivada da função de saída h ao longo de f isto é:

$$\begin{aligned}
 z_1 &= y = h(x) \\
 z_2 &= \dot{y} = L_f h(x) \\
 z_3 &= \ddot{y} = L_f^2 h(x) \\
 &\dots \\
 z_n &= y^{(n-1)} = L_f^{(n-1)} h(x)
 \end{aligned} \tag{39}$$

A função de linearização é obtida igualando-se a derivada de ordem n à nova entrada v , isto é:

$$y^{(n)} = L_f^{(n)} h(x) + L_g L_f^{n-1} h(x) u = v. \tag{40}$$

Isto corresponde a uma entrada u dada por:

$$u = \gamma(x, v) = \frac{1}{L_g L_f^{n-1} h(x)} (v - L_f^{(n)} h(x)) \tag{41}$$

A mudança de coordenadas é um mapeamento (difeomorfismo, veja apêndice II) $\phi: x \in X \rightarrow z \in X$, onde X é um conjunto aberto de \mathbb{R}^n , isto é:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \dots \\ z_n \end{pmatrix} = \phi(x) = \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \\ \dots \\ y^{(n-1)} \end{pmatrix} = \begin{pmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \\ \dots \\ L_f^{(n-1)} h(x) \end{pmatrix} \tag{42}$$

e a saída é dada por: $y = z_1$ (43)

O diagrama da figura 2 ilustra a aplicação da função de realimentação e a mudança de coordenadas.

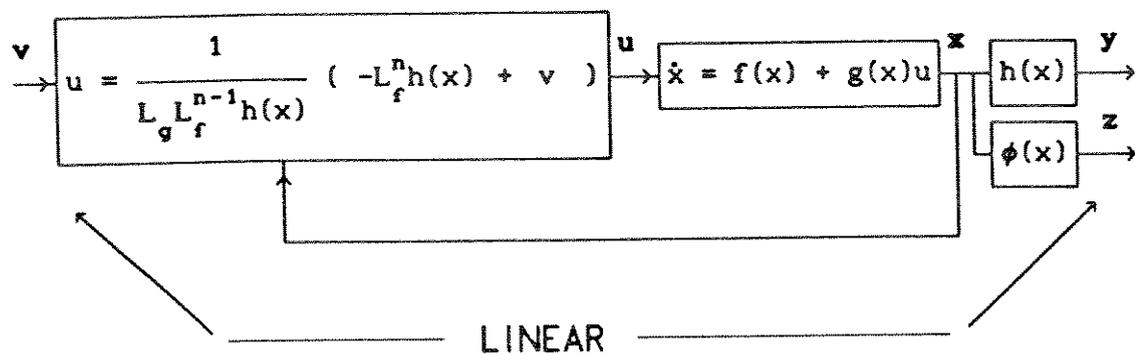


Fig. 2 - Linearização exata por realimentação de estado

A aplicação do mapeamento de linearização por realimentação de estado $\gamma(x,v)$ e a mudança de coordenadas $\phi(x)$ transforma o sistema não linear no seguinte sistema linear:

$$\begin{aligned}
 \dot{z}_1 &= z_2 \\
 \dot{z}_2 &= z_3 \\
 &\dots \\
 \dot{z}_{n-1} &= z_n \\
 \dot{z}_n &= v \\
 y &= z_1
 \end{aligned} \tag{44}$$

A aplicação do método exato de linearização por realimentação de estado, exige que o grau relativo seja igual à ordem do sistema (Isidori, 1989). Contudo, se o grau relativo não for igual a n , ainda assim será possível obter um sistema parcialmente linear, usando-se a própria função de saída h ou definindo-se uma outra função λ . Esse método é analisado a seguir.

Caso $r < n$:

No caso $r < n$, deve-se considerar duas diferentes soluções para o problema de linearização por realimentação. Em uma das soluções, em vez de se usar a própria função de saída do sistema, define-se uma outra função λ tal que, depois da mudança de coordenadas, a resposta entrada-estado será linear e na forma normal. O Lemma 2.4, página 161, em Isidori (1989) afirma que o problema de linearização exata no espaço de estado admite solução se existir uma vizinhança X de x^0 e uma função real $\lambda(x)$ definida em X , tal que o sistema,

$$\begin{cases} \dot{x} = f(x) + g(x) u \\ y = \lambda(x) \end{cases} \quad (45)$$

tenha grau relativo igual a n em torno de x^0 .

Para se determinar λ deve-se resolver um conjunto de equações diferenciais parciais dado por:

$$L_g \lambda(x) = L_{ad_{fg}} \lambda(x) = \dots = L_{ad_{fg}^{n-2}} \lambda(x) = 0; \quad (46)$$

veja a definição de $L_{ad_{fg}^n}$ no Apêndice II.

Segundo o Lemma 2.4 em Isidori (1989), para que o problema de linearização exata no espaço de estado tenha solução, ou seja, a condição de existência de uma função de saída $\lambda(x)$ para a qual o grau relativo do sistema seja igual a n em x^0 , corresponde às duas condições seguintes:

a. A matrix

$$\begin{bmatrix} g(x^0) & ad_{fg}(x^0) & \dots & ad_{fg}^{n-2}(x^0) & ad_{fg}^{n-1}(x^0) \end{bmatrix}$$

tenha posto igual a n .

b. A distribuição $D = \text{span}\{ g, \text{ad}_{fg} \dots \text{ad}_{fg}^{n-2} \}$ seja involutiva em uma proximidade X de x^0 (veja Apêndice II).

Essa solução é de pouco valor prático devido a dificuldade de se determinar a função λ . Contudo, é possível se obter um sistema parcialmente linear, isto é, linear com relação a resposta entrada-saída e onde um subconjunto das equações de estado sejam representadas por um sistema de equações lineares, na forma normal. Esta solução é discutida a seguir.

Neste caso a linearização por realimentação é realizada aplicando-se o mesmo método usado no caso $r=n$, exceto que, somente as primeiras r variáveis de estado são consideradas no processo de linearização. A mudança de coordenadas é obtida derivando-se a própria função de saída do sistema h , ao longo de f . Desta forma, são definidas as primeiras r variáveis de estado do modelo linear, isto é:

$$\begin{aligned} z_1 &= y = h(x) \\ z_2 &= \dot{y} = L_f h(x) \\ z_3 &= \ddot{y} = L_f^2 h(x) \\ &\dots \\ z_r &= y^{(r-1)} = L_f^{(r-1)} h(x) \end{aligned} \tag{47}$$

A entrada u é obtida igualando-se a derivada da saída de ordem r à entrada v ,

$$y^{(r)} = L_f^{(r)} h(x) + L_g L_f^{r-1} h(x) u = v \tag{48}$$

o que corresponde a aplicação da entrada u , determinada por realimentação de estado, isto é:

$$u = \gamma(x, v) = \frac{1}{L_g L_f^{r-1} h(x)} (v - L_f^{(r)} h(x)) \quad (49)$$

Se a mudança de coordenadas (50) for realizada após a aplicação da função de linearização por realimentação de estado (49), obtém-se um sistema parcialmente linear. Este sistema é linear em termos de z_1, \dots, z_r , mas não linear em termos das variáveis restantes z_{r+1}, \dots, z_n . A mudança de coordenadas se torna:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \dots \\ z_r \end{pmatrix} = \phi(x) = \begin{pmatrix} y \\ \dot{y} \\ \ddot{y} \\ \dots \\ y^{(r-1)} \end{pmatrix} = \begin{pmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \\ \dots \\ L_f^{(r-1)} h(x) \end{pmatrix} \quad (50)$$

Assim sendo, a aplicação do controle por realimentação de estado $\gamma(x, v)$ e da mudança de coordenadas $\phi(x)$ transforma o sistema não linear no seguinte sistema parcialmente linear:

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\dots \\ \dot{z}_{r-1} &= z_r \\ \dot{z}_r &= v \\ y &= z_1 \end{aligned} \quad (51)$$

Pode-se também escolher funções $\zeta_1, \dots, \zeta_{n-r}$ tais que as outras $n-r$ variáveis de estado definidas em (52) sejam independentes da entrada v . No novo sistema de coordenadas o

sistema será representado por:

$$\begin{aligned}
 \dot{z}_1 &= z_2 \\
 \dot{z}_2 &= z_3 \\
 &\dots \\
 \dot{z}_{r-1} &= z_r \\
 \dot{z}_r &= v \\
 \dot{z}_{r+1} &= \zeta_1(z) \\
 &\dots \\
 \dot{z}_n &= \zeta_{n-r}(z) \\
 y &= z_1
 \end{aligned} \tag{52}$$

Este é um sistema parcialmente linear, da entrada para a saída e da entrada para as primeiras r variáveis de estado. Pode-se observar também que as últimas $n-r$ variáveis não são influenciadas pela entrada v (veja seção 1.4, dinâmica nula). A figura 3 ilustra a aplicação da função de linearização e da mudança de coordenadas para o caso $r < n$.

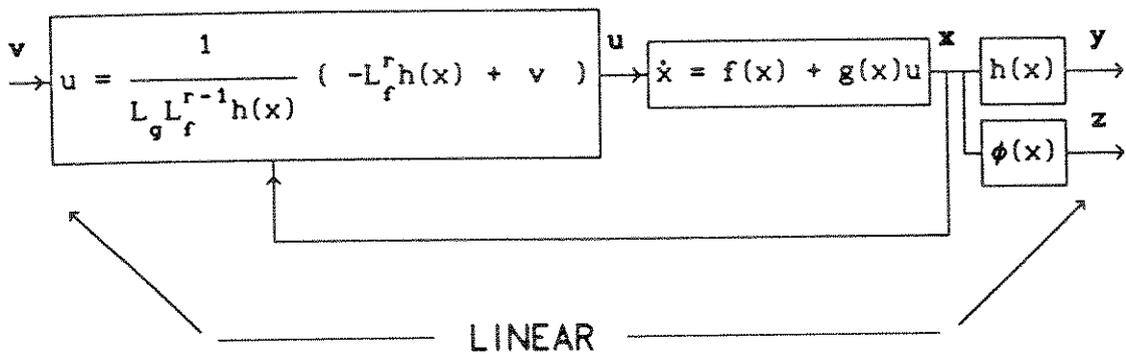


Fig. 3 - Linearização por realimentação no espaço de estado para $r < n$

Um outro conceito importante a ser considerado, é conhecido como dinâmica nula de num sistema não linear e será

apresentado a seguir.

1.4 Dinâmica Nula

O conceito de dinâmica nula, em sistemas não lineares (veja Isidori, 1989), é similar ao conceito de zeros em sistemas lineares. Em um sistema linear, o grau relativo r pode ser considerado como a diferença entre o número de polos e o número de zeros da função de transferência. Em particular, qualquer sistema linear com $r < n$ possui zeros finitos na função de transferência e se $r = n$, o sistema não tem zeros finitos.

Na equação (52) tem-se um sistema linear de dimensão $r < n$, em forma normal, acoplado a um sistema não linear de ordem $n - r$. Decompondo-se o vetor de estado z de acordo com:

$$\xi = \begin{pmatrix} z_1 \\ \dots \\ z_r \end{pmatrix} \quad (53)$$

$$e \quad \eta = \begin{pmatrix} z_{r+1} \\ \dots \\ z_n \end{pmatrix} \quad (54)$$

e se na equação (48), for aplicado:

$$L_f^{(r)} h(x) = L_f^{(r)} h(\phi^{-1}(z)) \quad (55)$$

$$\text{ou } L_f^{(r)} h(\phi^{-1}(z)) = L_f^{(r)} h(\phi^{-1}(\xi, \eta)) = b(\xi, \eta) \quad (56)$$

$$\begin{aligned}
e \quad L_g^{(r)} L_f^{r-1} h(x) &= L_g L_f^{(r-1)} h(\phi^{-1}(z)) = \\
&= L_g L_f^{(r-1)} h(\phi^{-1}(\xi, \eta)) = a(\xi, \eta)
\end{aligned} \tag{57}$$

então, pode-se escrever:

$$v = b(\xi, \eta) + a(\xi, \eta)u \tag{58}$$

e o conjunto de equações (52) poderá ser escrito como,

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= z_3 \\
&\dots \\
\dot{z}_{r-1} &= z_r \\
\dot{z}_r &= b(\xi, \eta) + a(\xi, \eta)u \\
\dot{\eta} &= q(\xi, \eta)
\end{aligned} \tag{59}$$

onde o mapeamento q corresponde a uma função em \mathbb{R}^{n-r} definida em um conjunto aberto X de \mathbb{R}^n . Da equação (52), pode-se observar que:

$$q(\xi, \eta) = q(z) = \begin{pmatrix} q_1(z) \\ \dots \\ q_{n-r}(z) \end{pmatrix} = \begin{pmatrix} \xi_1(z) \\ \dots \\ \xi_{n-r}(z) \end{pmatrix} \tag{60}$$

Considere que $x^\circ \in X$ seja um ponto de equilíbrio do sistema com $u^\circ \equiv 0$, isto é $f(x^\circ) = 0$ e $h(x^\circ) = 0$, então $\xi = 0$. Se $\eta=0$, $(\xi, \eta) = (0,0)$ for um ponto de equilíbrio do sistema nas novas coordenadas então $b(\xi, \eta) = 0$ e $q(\xi, \eta)=0$.

De uma forma mais geral, existe um estado inicial x° e uma função de entrada $u^\circ(t)$ (não necessariamente nula), tal que, nas proximidades de $t=0$, tem-se $y(t) = 0$, e desde que

$y(t) = z_1(t)$, $z_1(t)=0$. Isto implica em:

$$\dot{z}_1(t) = \dot{z}_2(t) = \dots = \dot{z}_r(t) = 0 \quad (61)$$

e $\xi(t) = 0$ nas vizinhanças de $t=0$. Neste caso, a entrada u será a solução de:

$$b(0,\eta) + a(0,\eta)u = 0 \quad (62)$$

e de (59), $\eta(t)$ será regido pela seguinte equação diferencial:

$$\dot{\eta} = q(0,\eta) \quad (63)$$

A dinâmica da equação (63) é conhecida como Dinâmica Nula, e este conceito é importante para a solução do problema de estabilização assintótica de um sistema não linear. Por outro lado, a dinâmica nula de um sistema não linear deverá ser necessariamente estável para aplicação do método de linearização por realimentação através do emprego de redes neurais, a ser apresentado no Cap. IV. Um dos motivos para esta restrição é que há necessidade de se coletar amostras de entrada e saída por longos períodos, o que seria impraticável em um sistema instável. Considera-se que um sistema não linear é de realização mínima quando sua dinâmica nula é assintoticamente estável.

2. Linearização por realimentação de sistemas de tempo contínuo de múltiplas entradas e múltiplas saídas (MIMO)

O método de linearização por realimentação, para sistemas do tipo ψ de múltiplas entradas e múltiplas saídas (eq. 1), é significativamente mais complexo que o caso de entrada e saída únicas. Mostra-se no capítulo IV que o uso de redes neurais pode ser também aplicado para sistemas MIMO. Nesta seção apresenta-se uma introdução ao caso de sistemas de tempo

contínuo. Para simplificar a análise, serão somente considerados sistemas onde o número de entradas é igual ao número de saídas (Isidori, 1989).

Para sistemas MIMO, o grau relativo é um conjunto de inteiros $\{r_1, \dots, r_m\}$, tais que no ponto x^0 :

$$(a) L_{g_j} L_f^k h_i(x) = 0 \quad (64)$$

para qualquer $1 \leq i, j \leq m$, $k < r_i - 1$, e para qualquer x nas proximidades de x^0 ,

(b) A matriz $m \times m$,

$$A(x) = \begin{pmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & \dots & L_{g_m} L_f^{r_2-1} h_2(x) \\ \dots & & \dots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{pmatrix} \quad (65)$$

seja não singular no ponto $x=x^0$. O grau relativo r_i , para um sistema MIMO, pode ser interpretado como o número de vezes que a saída i deve ser diferenciada para se ter pelo menos um dos componentes do vetor de entrada u aparecendo na expressão da derivada.

No caso MIMO, linearização de entrada-saída por realimentação consiste na aplicação de uma realimentação de estado,

$$u_i = \gamma(x, v) = \alpha_i + \sum_{j=1}^m \beta_{ij}(x) v_j \quad (66)$$

onde $v = \{v_1, \dots, v_m\}$ representa a entrada exógena, $\alpha_i(x)$ e $\beta_{ij}(x)$ para $1 \leq i, j \leq m$, são funções suaves definidas em um conjunto aberto X de \mathbb{R}^n . A figura (4) ilustra a aplicação da realimentação de estado.

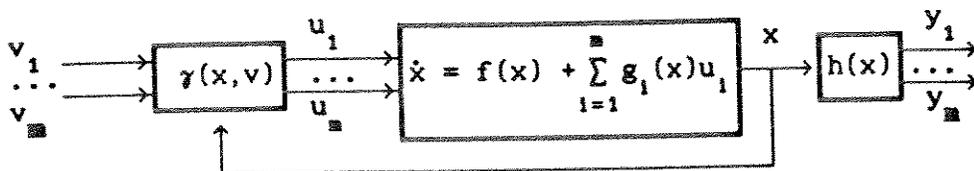


Fig. 4 - Linearização de entrada-saída por realimentação de estado em sistemas MIMO.

A aplicação da realimentação de estado (66) transforma o sistema ψ no seguinte sistema:

$$\begin{cases} \dot{x} = f(x) + \sum_{i=1}^m g_i(x)\alpha_i(x) + \sum_{i=1}^m \left(\sum_{j=1}^m g_j(x)\alpha_{ji}(x) \right) v_i & (67a) \end{cases}$$

$$\begin{cases} y = h(x) & (67b) \end{cases}$$

Se a correta mudança de coordenadas,

$$z = \phi(x) \quad (68)$$

for aplicada, o sistema será transformado em:

$$\begin{cases} \dot{z} = Az + Bv & (69) \end{cases}$$

$$\begin{cases} y_i = z_i, \quad i=1, \dots, m & (70) \end{cases}$$

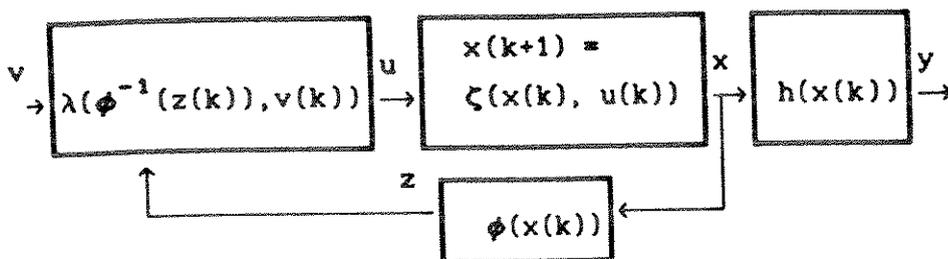


Fig. 7 - Aplicação da linearização por realimentação e da mudança de coordenadas.

O sistema da figura 7 pode ser linearizado de forma exata em termos da resposta entrada-estado, dependendo da escolha de λ e ϕ . Em alguns casos, não é possível obter um sistema completamente linear em termos de todo o estado do sistema, porém um sistema parcialmente linear pode ser obtido. Este é o caso onde o grau relativo é menor que a ordem do sistema (veja seção 3.2)

A Figura 8 mostra um outro caso onde a mudança de coordenadas não é aplicada, sendo o sistema linearizado somente em termos da resposta entrada-saída. Desta forma o sistema permanece não linear em termos da resposta entrada-estado.

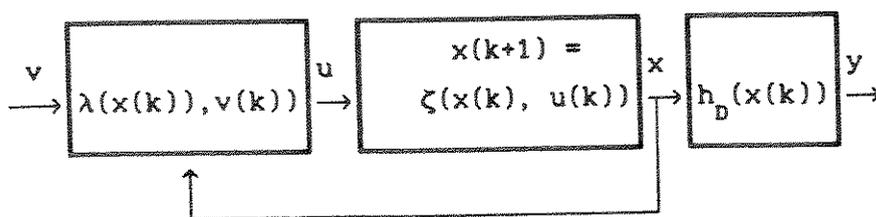


Fig. 8 - A aplicação da função de linearização sem a mudança de coordenadas.

As seguintes definições, propostas por Grizzle (1984), proporcionam os fundamentos para linearização de um sistema não linear de tempo discreto. A primeira definição se refere à

linearização global e a segunda, à linearização local. Inicialmente, é apresentada a definição de ponto de operação para sistemas tipo Σ_N .

Definição 1: Ponto de operação

Um par $(x(0), u(0)) \in \mathbb{R}^n \times \mathbb{R}^m$ é denominado ponto de operação do sistema $\Sigma : x(k+1) = \zeta(x(k), u(k))$ se,

$$\zeta(x(0), u(0)) = x(0) \quad (79)$$

Definição 2 - Linearização Global por Realimentação

O sistema $\Sigma : x(k+1) = \zeta(x(k), u(k))$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ é passível de linearização se existir uma função λ e um sistema de coordenadas globais z em \mathbb{R}^n e v em \mathbb{R}^m tais que,

$$\zeta(x, \lambda(z, v)) = A z + B v \quad (80)$$

para algumas matrizes constantes A e B .

Definição 3 - Linearização local por realimentação (Grizzle, 1986a)

O sistema $\Sigma : x(k+1) = \zeta(x(k), u(k))$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ é passível de linearização por realimentação, de forma local, nas proximidades do ponto de operação $(x(0), u(0))$ se:

a-existir um conjunto aberto $Z \subset X$ em torno de $x(0)$ e um conjunto aberto $V \subset U$ em torno de $u(0)$ tal que:

$$\zeta : Z \times U \rightarrow X \quad (81)$$

b-existir uma função de realimentação não singular $\lambda : Z \times V \rightarrow U$ e coordenadas locais z e v definidas em X e U tais que:

$$\zeta(x, \lambda(z, v)) = A z + B v \quad (82)$$

para qualquer $z \in Z$ e $v \in V$.

Das definições acima, conclui-se que o método de linearização por realimentação consiste, basicamente na determinação da função de realimentação λ e da mudança de coordenadas ϕ . Na próxima seção, serão apresentados os procedimentos para se determinar λ e ϕ para sistemas com equações conhecidas. No capítulo IV, será descrito um método aproximado para determinação de λ e ϕ , para sistemas não identificados, usando-se redes neurais.

3.2 Linearização por realimentação em sistemas de tempo discreto SISO tipo \sum_N

Para se estabelecer os procedimentos para linearização por realimentação, o conceito de grau relativo em sistemas de tempo discreto (El-Bially, 1990) será inicialmente apresentado.

Definição 4 - Grau Relativo de um sistema de tempo discreto:

O grau relativo r de um sistema SISO de tempo discreto tipo \sum_N é localmente definido como o número de intervalos de tempo r após os quais a entrada aparecerá explicitamente na expressão de saída.

Para isto, deverá haver uma vizinhança X em torno de $x(0)$ onde:

$$\begin{aligned} 1. \frac{\partial y(k+1)}{\partial u(k)} &= 0 \quad \text{para } i < r \\ 2. \frac{\partial y(k+1)}{\partial u(k)} &\neq 0 \quad \text{para } i = r \end{aligned} \quad (83)$$

Da mesma forma que no caso de sistemas de tempo contínuo, os sistemas de tempo discreto que possuem grau relativo igual a sua própria ordem ($r=n$), podem ser linearizados de forma exata, desde que satisfeitas as condições de linearização global. Quando o grau relativo for menor que n , pode-se obter um sistema parcialmente linear.

A aplicação dos seguintes procedimentos, incluindo a mudança de coordenadas, faz com que o sistema se transforme em um sistema linear na forma de Brunovsky. Quando $r < n$, o novo sistema se torna parcialmente linear, com somente as r primeiras equações de estado na forma normal de Brunovsky. As $n-r$ equações restantes se tornam não-lineares em relação ao estado e à entrada. Em certos casos, é possível se obter equações que sejam somente função do estado. Assim sendo:

$$\begin{aligned}
 \text{a- Para } \underline{r=n}: \quad z_1(k+1) &= z_2(k) \\
 z_2(k+1) &= z_3(k) \\
 &\dots \\
 z_{n-1}(k+1) &= z_n(k) \\
 z_n(k+1) &= v(k)
 \end{aligned}
 \tag{84}$$

b- Para $r < n$:

$$\begin{aligned}
z_1(k+1) &= z_2(k) \\
z_2(k+1) &= z_3(k) \\
&\dots \\
z_{r-1}(k+1) &= z_r(k) \\
z_r(k+1) &= v(k) \\
z_{r+1}(k+1) &= \zeta_1(z, v) \\
&\dots \\
z_n(k+1) &= \zeta_{n-r}(z, v)
\end{aligned}
\tag{85}$$

Desta forma, uma vez obtido o valor do grau relativo, pode-se então determinar a função de realimentação e a transformação de coordenadas necessárias para linearizar o sistema.

Para se estabelecer as expressões de ϕ e λ , considere uma mudança de coordenadas tal que $y(k) = z_1(k)$. Então, no caso geral $r \leq n$, os instantes subsequentes serão:

$$\begin{aligned}
y(k) &= h(x(k)) = z_1(k) \\
y(k+1) &= h(x(k+1)) = z_1(k+1) = z_2(k) \\
y(k+2) &= h(x(k+2)) = z_1(k+2) = z_2(k+1) = z_3(k) \\
&\dots \\
y(k+r-1) &= h(x(k+r-1)) = z_1(k+r-1) = z_2(k+r-2) = \dots = z_r(k) \\
y(k+r) &= h(x(k+r)) = z_1(k+r) = z_2(k+r-1) = z_3(k+r-2) = \dots = \\
&= z_r(k+r-(r-1)) = \\
&= z_r(k+1)) = \\
&= v(k)
\end{aligned}
\tag{86}$$

Assim sendo, a mudança de coordenadas fica:

$$z(k) = \begin{pmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \\ \dots \\ z_{r-1}(k) \\ z_r(k) \end{pmatrix} = \begin{pmatrix} \phi_1(x(k)) \\ \phi_2(x(k)) \\ \phi_3(x(k)) \\ \dots \\ \phi_{r-1}(x(k)) \\ \phi_r(x(k)) \end{pmatrix} = \begin{pmatrix} h(x(k)) \\ h(\zeta(x(k))) \\ h(\zeta(\zeta(x(k)))) \\ \dots \\ \overset{r-2 \text{ vezes}}{\widehat{\zeta(\zeta \dots (\zeta(x(k))))}} \\ \overset{r-1 \text{ vezes}}{\widehat{\zeta(\zeta \dots (\zeta(x(k))))}} \end{pmatrix} \quad (87)$$

A transformação acima não é função de $u(k)$, pois somente as r primeiras equações foram consideradas. Se a transformação for calculada para o passo seguinte, a entrada $u(k)$ aparecerá na expressão, i. e.,

$$h(x(k+r)) = v(k) = h(\overset{r \text{ vezes}}{\widehat{\zeta(\zeta \dots (\zeta(x(k), u(k))))}}) \quad (88)$$

Representando-se a transformação acima por:

$$v(k) = \lambda^{-1}(x(k), u(k)) \quad (89)$$

a função de realimentação ficará:

$$u(k) = \lambda(x(k), v(k)) \quad (90)$$

onde se assume que a função λ seja bijetiva.

Para o caso $r < n$, somente as r primeiras equações, representam um sistema linear, nas novas coordenadas, embora as

n-r equações no conjunto (85) possam ser escolhidas de forma a serem independentes da entrada, isto é:

$$\frac{\partial \phi_i(k)}{\partial u(k)} = 0 \quad i = r+1, \dots, n \quad (91)$$

O método exige também que a transformação global de coordenadas ϕ seja passível de inversão.

A aplicação da transformação de coordenadas (87) e da função de realimentação (90), transforma o sistema não linear em um sistema parcialmente linear, onde as primeiras r equações estão na forma normal de Brunovsky, como na equação (85). Na próxima seção, este método é estendido para sistemas da classe Σ , onde a entrada aparece de forma linear.

3.3 Linearização por Realimentação em Sistemas Não Lineares SISO de Tempo Discreto da Classe Σ

O mesmo procedimento, da seção anterior, pode ser aplicado para a classe Σ de sistemas não lineares, pois o conjunto de equações (86) se aplica também neste caso. Considerando-se que o produto $g(x(k))u(k)$ se anule nas primeiras r equações a transformação de coordenadas fica definida somente em termos de h e f, isto é:

$$z(k) = \begin{pmatrix} z_1(k) \\ z_2(k) \\ \dots \\ z_{r-1}(k) \\ z_r(k) \end{pmatrix} = \phi(k) = \begin{pmatrix} \phi_1(x(k)) \\ \phi_2(x(k)) \\ \dots \\ \phi_{r-1}(x(k)) \\ \phi_r(x(k)) \end{pmatrix} = \begin{pmatrix} h(x(k)) \\ h(f(x(k)) + \overbrace{g(x(k))u(k)}^{\hat{=}0}) \\ \dots \\ h(\overbrace{f(f \dots (f(x(k))))}^{r-2 \text{ vezes}}) \\ h(\overbrace{f(f \dots (f(x(k))))}^{r-1 \text{ vezes}}) \end{pmatrix} \quad (92)$$

Considerando-se que o grau relativo seja igual a r, se a transformação acima for aplicada para mais um passo, tem-se:

$$h(x(k+r)) = v(k) = h(\overbrace{f(f \dots (f(f(x(k)) + g(x(k))u(k))))}^{r-1 \text{ vezes}}) \quad (93)$$

$$\text{Considerando-se, } \phi_r(.) = h(\overbrace{f(f(f(.)))}^{r-1 \text{ vezes}}) \quad (94)$$

e substituindo-se em (93), obtém-se:

$$v(k) = \phi_r(f(x(k)) + g(x(k))u(k)) \quad (95)$$

e considerando-se também que ϕ_r seja passível de inversão, tem-se:

$$\phi_r^{-1}(v(k)) = f(x(k)) + g(x(k))u(k) \quad (96)$$

e a entrada u(k), capaz de linearizar o sistema pode, então, ser calculada por:

$$u(k) = (\phi_r^{-1}(v(k)) - f(x(k))) \frac{1}{g(x(k))} \quad (97)$$

A transformação de coordenadas (92) e a função (97) linearizam, por realimentação, o sistema de tempo discreto Σ , considerando-se que as condições de linearização globais ou locais, estabelecidas por Grizzle (1984), sejam satisfeltas (veja definições 2 e 3).

3.4 Linearização por Realimentação em Sistemas MIMO de Tempo Discretos

Sistemas de tempo discreto com múltiplas entradas e múltiplas saídas também podem ser linearizados, usando-se uma extensão do procedimento descrito para sistemas de entrada e saída únicas. Da mesma forma que no caso de sistemas contínuos, uma das diferenças básicas é que, sistemas MIMO possuem um vetor de graus relativos, que devem ser determinados para que se possa aplicar o método de linearização por realimentação. Monaco & Normand (1983a) mostraram que não somente a linearização por realimentação pode ser resolvida, mas também o problema de desacoplamento de entrada-saída, para sistemas com mesmas dimensões de entrada e saída. Considerando-se que, para o método proposto no capítulo IV, os detalhes do método analítico se tornam irrelevantes, pois o processo através de redes neurais se constitui em um esquema totalmente diferente, os detalhes matemáticos do método analítico não serão tratados neste trabalho.

Na seção seguinte, serão consideradas as implicações de se amostrar sistemas contínuos para se aplicar linearização por realimentação.

4. Efeitos do processo de amostragem em linearização por realimentação

Considerando-se que a maioria dos sistemas discretos se constitui em versões amostradas de sistemas contínuos, a aplicação da técnica de linearização por realimentação deve ser analisada para estes casos.

O principal objetivo deste trabalho é a linearização por realimentação através de redes neurais. Conforme apresentado no capítulo IV, o método é aplicável tanto para sistemas de tempo contínuo como para sistemas de tempo discreto ou amostrados. No caso de sistemas de tempo contínuo a rede neural poderia também ser treinada através de amostragem, em um esquema off-line, e ser depois implementada usando-se dispositivos analógicos. Entretanto a tecnologia atual de redes neurais não permite implementações deste tipo. Conforme será mostrado através dos exemplos simulados, mesmo para o caso de sistemas amostrados, o esquema apresentado no capítulo IV, apresenta resultados bastante satisfatórios. Isto se deve ao fato de que, através do método de identificação usando redes neurais, não se identifica a função de linearização analítica mas a função capaz de fazer com que o sistema não linear apresente o mesmo desempenho do sistema linear, quando o mesmo período de amostragem, usado para treinamento da rede, for também aplicado durante a operação normal.

Por outro lado, a aplicação de amostragem nos esquemas convencionais de linearização por realimentação tem sido objeto de várias críticas por parte de diversos pesquisadores. Sontag(1983,1986a,1986b) analisou questões referentes a controlabilidade e observabilidade de sistemas não lineares amostrados. Jakubczyk(1987) investigou especificamente os efeitos da aplicação de amostragem para a implementação de linearização

por realimentação e mostrou que restrições bastante severas são introduzidas na controlabilidade do sistema original.

Considere o sistema contínuo ψ (eq. II.01) definido localmente em torno de x_0, u_0 , e onde os campos vetoriais f e g são completos na vizinhança de x_0, u_0 . Suponha que o sistema de tempo discreto Σ_N (eq. II.16) seja identificado amostrando-se ψ no intervalo de tempo $\delta \neq 0$ e onde $\zeta: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ seja um mapeamento que satisfaça a solução $x(t+\delta)$ de ψ com condições iniciais $x(t)$ e controle constante $u(t)$ no intervalo $[t, t+\delta]$. Pode-se verificar que,

$$\begin{aligned} x(t+\delta) &= x(k+1) = \zeta(x(k), u(k)) = \\ &= \exp((f(x(t)) + u_1(t)g_1(x(t)) + \dots + u_m(t)g_m(x(t)))\delta) \end{aligned} \quad (98)$$

Jakubczyk(1987) mostrou que ψ é linearizável por realimentação amostrada se existir um subconjunto aberto $\Delta \subseteq \mathbb{R}_+$ tal que ψ seja linearizável para cada $\delta \in \Delta$. Pode-se também interpretar o conceito de linearização por realimentação amostrada considerando-se que ψ possa ser transformado em um sistema linear f (eq. II.10) no tempo de amostragem $\delta, 2\delta, \dots$, por meio da aplicação de uma mudança de coordenadas e realimentação de estado implementada através de segurador de ordem zero. Se o sistema não linear de tempo contínuo for linearizável por realimentação amostrada, a saída deverá ser igual ao saída do sistema linear f , a cada instante de amostragem. Conforme se pode observar nos exemplos processados no capítulo IV, esta condição foi atingida com bastante precisão, através do processo de linearização por realimentação usando-se redes neurais.

Entretanto, a abrangência da aplicação da linearização por realimentação amostrada ainda está sendo investigada por vários pesquisadores. Grizzle(1986a) afirmou que o método de linearização por realimentação amostrada é tão restritivo que

implicaria em que o sistema contínuo não linear original fosse, de fato, um sistema linear. Jakubczyk (1987) discorda desta afirmação mas conclui que "para sistema contínuos não lineares onde $n > 1$, a linearização por realimentação amostrada, realmente introduz restrições bastante severas no sistema e que a classe de sistemas linearizáveis por realimentação amostrada se constitui em um subconjunto pequeno da classe de sistemas linearizáveis por realimentação".

Todas estas afirmações se referem a implementação da linearização por realimentação através de funções determinadas analiticamente. No processo de linearização por redes neurais proposto no capítulo IV, a função de linearização é estimada para aplicação com o mesmo intervalo de amostragem usado para treinamento. Desta forma, a rede é treinada a atuar de maneira tal que o sistema não linear tenha o mesmo desempenho do sistema linear a cada instante de amostragem. Assim sendo, caso seja necessário mudar o intervalo de amostragem, novo treinamento deverá ser realizado. Portanto, as considerações restritivas apresentadas por Jakubczyk e Grizzle não se aplicam para o caso de estimação da função de linearização por realimentação através de redes neurais, proposto no capítulo IV. Esta conclusão obtida de forma qualitativa neste capítulo e mostrada através das simulações no capítulo IV, carece entretanto de uma demonstração matemática mais rigorosa e, portanto, deverá fazer parte de futuras investigações.

5. Conclusões

Neste capítulo se discutiu o método analítico de linearização por realimentação para sistemas de tempo discreto e contínuo. O principal motivo desta análise, consiste em se estabelecer uma estrutura analítica para servir de base de comparação com o método desenvolvido no capítulo IV. Para se aplicar os métodos aqui revistos, torna-se necessário que as equações do sistema sejam conhecidas. O método apresentado no capítulo IV propõe que uma determinada classe de sistemas não lineares, não identificados, de tempo discreto ou contínuo, sejam linearizados usando-se redes neurais de várias camadas treinadas através do algoritmo de propagação retroativa.

CAPITULO III

DINÂMICA INVERSA DE SISTEMAS NÃO LINEARES USANDO REDES NEURAIS

Apresenta-se neste capítulo, um método para se identificar a dinâmica inversa de sistemas não lineares por meio de redes neurais. São também apresentados resultados de simulações, onde se identifica o mapeamento inverso de uma determinada classe de sistemas não lineares, passíveis de inversão, usando-se dados de entrada-saída e uma rede neural.

O problema de se construir a dinâmica inversa de um sistema para se recuperar o sinal de entrada saída, foi inicialmente estudado por, Silverman (1969) e Sain (1969), para sistemas lineares. Condições suficientes para inversão de uma determinada classe de sistemas não lineares foi apresentada por Hirschorn (1979), onde se considera que as condições iniciais do sistema sejam conhecidas. Singh (1982) também apresentou condições suficientes para inversão de sistemas multivariáveis não-lineares e observáveis, baseando-se no algoritmo de Hirschorn, exceto que a questão de inversão com condições iniciais desconhecidas foi também considerado. Rephurn (1980), apresenta um critério diferente para inversão de sistemas multivariáveis de entrada-saída da classe C^∞ . Neste capítulo, apresenta-se um método para identificar um mapeamento estático e aproximado da função inversa, para sistemas passíveis de inversão, usando-se redes neurais de várias camadas e baseado em dados de entrada-saída do sistema. Inicialmente, são revistas as condições de inversão baseadas no trabalho de Hirschorn.

1. Método de Inversão de Hirschorn

O método desenvolvido por Hirschorn (1979) para sistemas não lineares é uma extensão do processo de Silverman (1969) para sistemas lineares. Uma sequência de sistemas é construída mudando-se as funções de saída até que se possa resolver o sistema de equações, determinando-se a entrada u em termos da variável de saída y , de sua derivada e da variável de estado x . Este processo resulta em um segundo sistema que atua como um inverso à esquerda do sistema original.

Determinando-se a primeira derivada da saída, para o sistema ψ (eq. II.1), tem-se:

$$\begin{aligned} y^{(1)} &= \frac{dh}{dx} \frac{dx}{dt} = \frac{dh}{dx} \left(f(x) + \sum_{i=1}^m u_i g_i(x) \right) = \\ &= L_f h(x) + \sum_{i=1}^m u_i L_{g_i} h(x) \end{aligned} \quad (01)$$

$$\text{Considere } u \in \mathbb{R}^m = [u_1, \dots, u_m]^T \quad (02)$$

$$\text{e } D(x) = [L_{g_1} h(x) \quad L_{g_2} h(x) \quad \dots \quad L_{g_m} h(x)] \quad (03)$$

uma matriz $p \times m$ para cada $x \in M$, onde M é um manifold analítico em \mathbb{R}^n . Considere também,

$$f_h(x) = L_f h(x), \quad (04)$$

a derivada Lie de h ao longo de f .

A derivada da saída se torna:

onde M_k é um submanifold denso de M , $C_k(x)$ é um campo vetorial (veja apêndice II) de dimensões $p \times 1$ e $D_k(x)$ é uma matriz $p \times m$, cujos elementos são funções analíticas reais em M_k , tais que:

$$D_k(x) = \begin{bmatrix} D_{k1}(x) \\ 0 \end{bmatrix} \quad (19)$$

e sendo $D_{k1}(x)$ uma matriz $r_k \times m$ de posto r_k para qualquer $x \in M_k$.

Para se construir o sistema ψ_{k+1} , a saída do sistema ψ_k deve ser decomposta em \bar{z}_k e \hat{z}_k , tal que,

$$z_k = \begin{bmatrix} \bar{z}_k \\ \hat{z}_k \end{bmatrix} = \begin{bmatrix} \bar{c}_k(x) \\ \hat{c}_k(x) \end{bmatrix} + \begin{bmatrix} D_{k1}(x) \\ 0 \end{bmatrix} u \quad (20)$$

e diferenciando-se \hat{z}_k com respeito a t ,

$$\hat{z}_k^{(1)}(t) = \frac{d\hat{c}_k}{dx} \left(f(x) + \sum_{i=1}^m u_i g_i(x) \right) \quad (21)$$

ou,
$$\hat{z}_k^{(1)}(t) = L_f \hat{c}_k(x) + \hat{D}_k(x) u \quad (22)$$

onde
$$\hat{D}_k(x) = \left(L_{g_1} \hat{c}_k(x), \dots, L_{g_m} \hat{c}_k(x) \right) \quad (23)$$

Então,
$$\begin{bmatrix} \bar{z}_k \\ \hat{z}_k^{(1)} \end{bmatrix} = \begin{bmatrix} \bar{c}_k(x) \\ L_f \hat{c}_k(x) \end{bmatrix} + \bar{D}_k(x) u \quad (24)$$

$$y^{(1)} = f_h(x) + D(x)u. \quad (05)$$

Seja $r_1 = \max_{x \in M} \{\text{posto } D(x)\}$ (06)

e seja E_o^1 uma matriz elementar que reordena as linhas de $D(x)$ de forma que as primeiras r_1 linhas sejam linearmente independentes, para qualquer $x \in M$, tal que:

$$E_o^1 D(x) = \begin{bmatrix} D_{11}(x) \\ D_{12}(x) \end{bmatrix} \quad (07)$$

onde $D_{11}(x)$ é uma matriz $r_1 \times m$ de posto r_1 .

Seja, $M_1 = \{x \in M \mid \text{rank } D_{11}(x) = r_1\}$ (08)

um *submanifold* denso e aberto de M (veja Hirschorn, 1979).

Desde que as primeiras r_1 linhas de $D(x)$ sejam linearmente independentes, pode-se realizar uma redução de linhas em $E_o^1 D(x)$. Para $x \in M$, define-se então uma matriz elementar $p \times q$:

$$E_o^2(x) = \begin{bmatrix} I_{r_1 \times r_1} & 0 \\ F_o(x) & I_{(q-r_1) \times (p-r_1)} \end{bmatrix} \quad (09)$$

onde a submatriz $F_o(x)$ é uma matriz $(p-r_1) \times r_1$ composta de funções analíticas em M_1 com a seguinte propriedade:

$$E_o^2(x) E_o^1 D(x) = \begin{bmatrix} D_{11}(x) \\ 0 \end{bmatrix}, \text{ para } x \in M_1 \quad (10)$$

e onde $D_{11}(x)$ é uma matriz $r_1 \times m$ de ranque r_1 para qualquer $x \in M_1$. Se a equação (05) for multiplicada por $E_0^2(x)E_0^1$ obtém-se:

$$E_0^2(x)E_0^1 \frac{dy}{dt} = E_0^2(x)E_0^1 L_f h(x) + E_0^2(x)E_0^1 D(x) u \quad (11)$$

Com base em (11) pode-se definir um outro sistema linear,

$$\text{Sistema } \psi_1: \begin{cases} \dot{x} = f(x) + \sum_{i=1}^m u_i g_i(x) \end{cases} \quad (12)$$

$$\begin{cases} z_1 = C_1(x) + D_1(x)u, \quad x \in M_1 \end{cases} \quad (13)$$

onde:

$$z_1(x) = E_0^2(x)E_0^1 \frac{dy}{dt} \quad (14)$$

$$C_1(x) = E_0^2(x)E_0^1 L_f h(x) \quad (15)$$

$$D_1(x) = E_0^2(x)E_0^1 D(x) = \begin{bmatrix} D_{11}(x) \\ 0 \end{bmatrix}, \text{ para } x \in M_1 \quad (16)$$

O número inteiro r_1 é denominado índice de inversão do sistema ψ_1 . O mesmo procedimento pode ser aplicado sequencialmente, definindo-se o sistema ψ_k , isto é:

$$\text{Sistema } \psi_k: \begin{cases} \dot{x} = f(x) + \sum_{i=1}^m u_i g_i(x) \end{cases} \quad (17)$$

$$\begin{cases} z_k = C_k(x) + D_k(x)u, \quad x \in M_k \end{cases} \quad (18)$$

e $D_{k+1,1}(x)$, sendo uma matriz $r_{k+1} \times m$ de ranque r_{k+1} para qualquer $x \in M_{k+1}$.

A matriz $D_{k+1}(x)$, como no caso do sistema ψ_1 , é definida de tal forma que:

$$E_k^2(x)E_k^1 \bar{D}_k(x) = \begin{bmatrix} \hat{D}_{k1}(x) \\ \hat{D}_{k2}(x) + F_k(x)\hat{D}_{k1}(x) \end{bmatrix} = \begin{bmatrix} \hat{D}_{k1}(x) \\ 0 \end{bmatrix} \quad (32)$$

Portando, para satisfazer (32), deve-se definir $F_k(x)$ tal que,

$$\hat{D}_{k2}(x) + F_k(x)\hat{D}_{k1}(x) = 0, \text{ para todo } x \in M_{k+1} \quad (33)$$

Se o mesmo procedimento for aplicado várias vezes, obtém-se uma sequência de inteiros positivos, r_1, r_2, r_3, \dots , onde r_k é denominado índice de inversão do sistema de ordem k . Por construção,

$$r_1 \leq r_2 \leq r_3 \leq \dots \leq m \quad (34)$$

Define-se ordem relativa α do sistema ψ (diferente de grau relativo definido no capítulo II), como o menor número inteiro positivo k , tal que $r_k = m$. Se $r_k < m$ para $k = 1, 2, \dots, \infty$, então $\alpha = \infty$. O inteiro α corresponde à ordem da maior derivada da saída $y(\cdot)$ necessária para acionar o sistema inverso à esquerda. Silverman (1969) mostrou que para sistemas lineares invariantes no tempo, $\alpha < \infty$ é a condição suficiente de inversibilidade do sistema. Para o caso não linear, $\alpha < \infty$ é somente uma condição necessária de inversibilidade do sistema ψ_α , e portanto, para se derivar o sistema inverso de ψ , há necessidade de se satisfazer outras condições adicionais (veja Hirschorn 1979).

Para se determinar a sequência de sistemas tem-se que resolver (33) em $F_k(x)$ e obter uma sequência de matrizes $F_0(x), F_1(x), F_2(x), \dots$, que, no caso linear, são constantes.

Em Hirschorn (1979), teorema 2, mostra-se que o sistema ψ_k é fortemente passível de inversão em x_0 , para qualquer $x_0 \in M_\alpha$ se, para todo $i \in \{1, 2, \dots, m\}$,

$$g_i \circ f^j \circ F_k(\cdot) = 0 \text{ on } M \quad (35)$$

para $0 \leq k \leq \alpha - 2$ e $0 \leq j \leq \alpha - 2 - k$, $f^j = f \circ f \circ \dots \circ f$, j vezes, e onde \circ representa composição de mapeamentos.

Para se descrever a construção do sistema inverso à esquerda do sistema não linear ψ , suponha que as condições descritas acima sejam satisfeitas e considere a ordem relativa do sistema $\alpha < \infty$. Então o sistema de ordem α terá a forma:

$$\dot{x} = f(x) + \sum_{i=1}^m u_i g_i(x) \quad (36)$$

$$z_\alpha = C_\alpha(x) + D_\alpha(x)u, \quad x \in M_\alpha \quad (37)$$

onde
$$D_\alpha(x) = \begin{bmatrix} D_{\alpha 1}(x) \\ 0 \end{bmatrix} \quad (38)$$

e $D_{\alpha 1}(x)$ é uma matriz $r_\alpha \times m$ de ranque r_α . Desde que $\alpha < \infty$, então tem-se $r_\alpha = m$, e $D_{\alpha 1}(x)$ será uma matriz $m \times m$ passível de inversão. Seja \bar{z}_α e \bar{c}_α os primeiros m componentes de z_α e C_α , respectivamente. Então,

$$\bar{z}_\alpha = \bar{c}_\alpha(x) + D_{\alpha 1}(x)u \quad (39)$$

onde $D_{\alpha_1}(x)$ é uma matriz $m \times m$ passível de inversão, para qualquer $x \in M_{\alpha}$. Existe também uma matriz $H_{\alpha}(x)$, de dimensão $m \times \alpha_p$, cujas entradas são funções analíticas reais de M_{α} , tais que,

$$\bar{z}_{\alpha} = H_{\alpha}(x) \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(\alpha)} \end{bmatrix} \quad (40)$$

e sendo assim, o seguinte sistema atuará como inverso a esquerda do sistema ψ :

$$\left\{ \begin{array}{l} \frac{d\hat{x}}{dt} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})\hat{u}; \end{array} \right. \quad (41)$$

$$\left\{ \begin{array}{l} \hat{y} = \hat{c}(\hat{x}) + \hat{D}(\hat{x})\hat{u}, \hat{x}_0 = x_0 \in M_{\alpha}, \end{array} \right. \quad (42)$$

onde:

$$\hat{f}(\hat{x}) = f(\hat{x}) - \left(g_1(\hat{x}) \dots g_m(\hat{x}) \right) D_{\alpha_1}^{-1}(\hat{x}) \bar{c}_{\alpha}(\hat{x}) \quad (43)$$

$$\hat{g}(\hat{x}) = \left(g_1(\hat{x}) \dots g_m(\hat{x}) \right) D_{\alpha_1}^{-1}(\hat{x}) H_{\alpha}(\hat{x}) \quad (44)$$

$$\hat{c}(\hat{x}) = - D_{\alpha_1}^{-1}(\hat{x}) \bar{c}_{\alpha}(\hat{x}) \quad (45)$$

$$\hat{D}(\hat{x}) = - D_{\alpha_1}^{-1}(\hat{x}) H_{\alpha}(\hat{x}) \quad (46)$$

Em um caso particular onde,

$$\hat{u}(t) = \begin{bmatrix} y^{(1)}(t) \\ \dots \\ y^{(\alpha)}(t) \end{bmatrix}, \quad (47)$$

tem-se, $\hat{y}(t) = u(t)$, (48)

isto é, a entrada do sistema ψ .

Uma conclusão interessante que pode ser tirada da aplicação deste procedimento para o caso linear (Hirschorn, 1979, remark 2), é que para sistemas lineares invariantes no tempo, os *manifolds* M, M_1, M_2, \dots correspondem ao espaço linear completo \mathbb{R}^n . Note também que para se recuperar a entrada u para o sistema ψ as equações dinâmicas (41) e (42) devem ser resolvidas, tendo como entrada um vetor composto das derivadas da saída $y^{(1)}, \dots, y^{(\alpha)}$, onde α é a ordem relativa do sistema. Deve-se notar também que é necessário que as condições iniciais do sistema sejam as mesmas condições iniciais do sistema original ψ . Considerando-se a dificuldade de se determinar $\hat{f}, \hat{g}, \hat{c}, \hat{D}$ e a necessidade de se conhecer as condições iniciais do sistema original, propõem-se na seção seguinte um método aproximado de se determinar o sistema inverso, usando-se redes neurais.

2. Dinâmica inversa de sistemas contínuos não lineares usando-se redes neurais.

O método descrito a seguir considera que a entrada de um sistema não linear pode ser recuperada a partir de, no mínimo, n_0 derivadas da saída. O número inteiro n_0 é definido por Singh (1982) como o índice de observabilidade, i. e., a derivada da saída de menor ordem tal que o conhecimento de $y, y^{(1)}, \dots, y^{(n_0)}$ seja suficiente para se determinar de forma única o estado $x(t) \in M$. Singh (1982) também sugere que o sistema inverso pode ser determinado para $n_0 \neq \alpha$, isto é, n_0 não necessita ser igual à ordem relativa do sistema definida por Hirschorn (1979). O método

que se propõe, nesta seção, se baseia no índice de observabilidade n_o , embora se proponha também um método aproximado de se determinar n_o , através do mesmo procedimento com redes neurais. Basicamente, somente sistemas que possam ser colocados na forma ψ_y (eq. II.9)¹ são tratados nesta seção. Assim sendo, neste caso tem-se, $n_o = n$, pois o sistema ψ_y poderá sempre ser escrito na forma ψ_N (eq. II.8).

As proposições seguintes estabelecem os conceitos básicos para se inverter os sistemas ψ , ψ_N ou ψ_y , usando-se redes neurais de múltiplas camadas.

Proposição 1: Uma rede neural pode ser treinada para aproximar a dinâmica inversa do sistema ψ (eq. II.1) baseada no estado e em suas derivadas se posto($g(x)$) = m , para qualquer $x \in X$, onde X é um subconjunto aberto de \mathbb{R}^n .

Demonstração: Supondo-se que o estado e sua derivada sejam mensuráveis e usando-se a seguinte notação:

$$u = (u_1, \dots, u_m)^T \quad (49)$$

$$e \quad g(x) = (g_1(x), \dots, g_m(x)) \quad (50)$$

A equação de estado do sistema ψ se torna:

$$\dot{x} = f(x) + g(x)u \quad (51)$$

Se posto($g(x)$) = m para qualquer $x \in X$, a equação (51) poderá ser resolvida para permitir que o controle seja calculado por:

$$u = (g^T(x)g(x))^{-1}g^T(x)(\dot{x} - f(x)), \quad x \in M \quad (52)$$

1 - Refere-se à equação 9 do capítulo II.

Isto significa que para se obter uma aproximação do mapeamento inverso torna-se necessário o conhecimento de amostras de u , x e \dot{x} .

Uma rede neural pode ser treinada para aproximar o mapeamento não linear representado pela equação (52) se uma quantidade suficiente de amostras for apresentada a rede, na fase de treinamento, de acordo com as condições de treinabilidade estabelecidas pelo teorema 1, capítulo 1. As amostras incluem dados de $(x, \dot{x}) \in X \times X_1$ e $u \in U$, onde X_1 é um subconjunto de X , e assumindo-se que f e g sejam funções suaves. ■

Proposição 2: Uma rede neural pode ser usada para aproximar um mapeamento inverso do sistema ψ_y (eq. II.9), isto é:

$$u = \theta^{-1}(y, y^{(1)}, y^{(2)}, \dots, y^{(n)})$$

se treinada com amostras dos seguintes dados:

-entrada: $y, y^{(1)}, y^{(2)}, \dots, y^{(n)}$
 -saída: $u(t)$

onde n é a ordem do sistema definida em (II.9).

Demonstração: Considere que θ seja um mapeamento suave e que o número de amostras e o intervalo de amostragem sejam escolhidos de forma a satisfazer a tolerância desejada. Pelo teorema de Hirschorn, para se obter o mapeamento inverso, α derivadas da saída devem ser fornecidas, sendo α a ordem relativa do sistema. Para se escrever as equações na forma II.9, pelo menos α derivadas

devem ser fornecidas. Assim sendo $n \geq \alpha$ é a condição suficiente para se estimar a entrada u de forma única. Para se treinar a rede neural é suficiente se aplicar as condições do teorema 1, (Cap. I). ■

Baseado na proposição 2, pode-se amostrar a saída do sistema não linear e uma rede neural pode ser treinada para captar a dinâmica inversa do sistema. Para sistemas do tipo ψ , quando o estado e sua derivada são conhecidos, a proposição 1 pode ser usada. Por outro lado, se as derivadas não são conhecidas, a proposição 2 pode ser usada para sistemas do tipo ψ ou ψ_N , pois se constituem casos especiais do sistema ψ_y . Os procedimentos descritos no capítulo I com relação ao dimensionamento de redes neurais podem ser diretamente aplicados neste caso. Muitas vezes, a estrutura das equações do sistema não estão identificadas e portanto, não se conhecem α ou n . Neste caso, o seguinte procedimento pode ser aplicado para se estimar a dinâmica inversa usando-se redes neurais.

2.1 Procedimento para identificação da dinâmica inversa usando-se redes neurais

Quando as equações do sistema e a ordem relativa α são conhecidas pode-se obter um mapeamento inverso exato, aplicando-se os métodos de Hirschorn, de Singh ou de Hephurn. Considerando-se que o procedimento por redes neurais é um método aproximado, e considerando-se que a função inversa é suave, pode-se obter um mapeamento inverso aproximado, mesmo se as derivadas de ordem mais elevadas não forem consideradas. Se muitas derivadas estiverem faltando, a tolerância desejada não será atingida durante a fase de treinamento da rede neural. Assim sendo, sugere-se o seguinte procedimento para sistemas não identificados:

Passo 1 - Escolher: um pequeno valor para n (baseado no conhecimento disponível sobre o sistema), um intervalo de amostragem e o número de amostras (baseado nos procedimentos estabelecidos no capítulo I).

Passo 2 - Excitar o sistema com valores aleatórios de entrada u e medir os valores correspondentes de saída e de suas derivadas, $y, y^{(1)}, \dots, y^{(n)}$.

Passo 3 - Iniciar o Procedimento de treinamento usando as diretivas do capítulo 1. A seguinte condição poderá ocorrer: os ajustes da rede neural poderão ficar presos em uma situação de mínimo local, no espaço de parâmetros, durante o processo de treinamento, mesmo que todas as condições sugeridas no capítulo I sejam satisfeitas. Considerando-se que o número de nós da camada oculta seja suficiente, a razão de ocorrência do mínimo local poderá ser o fato de que um número insuficiente de derivadas estejam sendo fornecidas. Portanto, o número de derivadas deverá ser aumentado e o processo deverá ser repetido, começando-se do Passo 1, até que a tolerância final desejada seja atingida. ■

Na realidade, não somente o mapeamento inverso é estimado, para o caso de sistemas não identificados do tipo ψ_y , mas também a ordem do sistema n . O valor estimado de n está relacionado com a tolerância de treinamento, o intervalo de amostragem e o alcance do espaço usado para treinamento.

Nota-se que, a amostragem está sendo aplicada somente para efeito de treinamento da rede neural. Após o treinamento, o mapeamento inverso aproximado poderá ser usado para qualquer esquema envolvendo sistemas contínuos, amostrados ou não. Assim

sendo, se o mapeamento inverso implementado por redes neurais, for usado para linearização por realimentação (veja Cap. II), as restrições apresentadas no capítulo II não se aplicam. Na seção seguinte, sistemas amostrados são analisados.

2.2 Obtenção do sistema inverso aproximado usando-se valores de saída amostrados

Considerando-se a dificuldade de se medir derivadas da saída em um sistema físico real, propõe-se nesta seção, um método aproximado de se estimar a dinâmica inversa. O método consiste em se armazenar valores instantâneos em vez de derivadas da saída. Esses valores passados da saída são usados como entrada para a rede neural. Certamente, o desempenho da dinâmica inversa, estimada desta forma, será uma função do período de amostragem δ . Consequentemente, para se usar este tipo de implementação de rede neural, o período de amostragem δ deverá ser mantido, após o treinamento. O sistema ψ_s , definido a seguir é uma versão amostrada do sistema ψ_y .

Sistema ψ_s :

$$y((k+1)\delta) = \theta(y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta), u(k\delta)) \quad (53)$$

onde θ é uma função suave, δ é o período de amostragem e n é a ordem do sistema amostrado. A saída y é definida como sendo um elemento do conjunto aberto Y de \mathbb{R}^p e u é a entrada, definida em um subconjunto aberto U de \mathbb{R}^m .

Definição 1: A função inversa θ^{-1} é um mapeamento de Y^{n+1} em U , tal que,

$$u(k\delta) = \theta^{-1}(y((k+1)\delta), y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta)) \quad (54)$$

A função inversa do sistema contínuo (amostrado) ψ_s fornece os valores $u(k\delta)$, constantes no intervalo δ (como em um segurador de ordem zero), capazes de acionar o sistema ψ_s , a partir dos seguintes valores prévios de saída $y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta)$, para o valor $y((k+1)\delta)$. Neste caso o valor inteiro n é o menor número de valores prévios de saída, necessários para se determinar $u(k\delta)$ de forma única. Supõe-se que θ^{-1} possa ser aproximada por uma rede neural de várias camadas sob tolerância τ , quando treinada com um número suficiente de amostras representando os espaços de definição Y^{n+1} e U .

Note que para um mesmo sistema, ψ_y e ψ_s são diferentes representações. Contudo, $y(k\delta)$ definida por ψ_s aproxima, ou se constitui, a versão amostrada de $y(t)$, definida por ψ_y , para $t = k\delta$ e para $u(t) = u(k\delta)$, constante no intervalo de amostragem δ .

Proposição 3: Uma rede neural pode aproximar o mapeamento inverso amostrado θ^{-1} (eq. 54) quando treinada com um número suficiente de amostras de:

$$u(k\delta) \in U, y((k+1)\delta), y(k\delta), \dots, y((k-n+1)\delta) \in Y^{n+1}.$$

Demonstração: Considerando-se que θ^{-1} seja uma função suave, unicamente definida por $y((k+1)\delta), y(k\delta), \dots, y((k-n+1)\delta) \in Y^{n+1}$, as condições suficientes para treinar uma rede neural de várias camadas que aproxima θ^{-1} são dadas pelo teorema 1, capítulo I, considerando-se que seja fornecido um número suficiente de amostras. ■

A proposição 3 fornece embasamento para se treinar uma

rede neural com a versão amostrada da dinâmica inversa do sistema. Se uma entrada (controle) u , constante no intervalo δ , for aplicada ao sistema ψ ou ψ_N , e suas saídas forem medidas, o mesmo procedimento aplicado para o sistema do tipo ψ_y poderá ser usado para aproximar a dinâmica inversa, considerando-se que ψ é uma subclasse de ψ_N e que ψ_N pode sempre ser escrito na forma ψ_y . Na última afirmação, supõe-se que ζ seja suave (eq. II.8).

No seguinte exemplo, a dinâmica inversa de um motor elétrico de corrente contínua (motor CC) com carga não linear é estimada usando-se redes neurais.

Exemplo 1: Neste exemplo a dinâmica inversa de um motor de corrente contínua é identificada usando-se redes neurais. As equações do sistema são as seguintes (em unidades normalizadas):

$$\frac{d\omega}{dt} = 1 + \lambda(\omega) \quad (55a)$$

$$\mu \frac{d\omega}{dt} = -\omega - i + u, \quad \mu = T_e / T_m \quad (55b)$$

onde:

ω - velocidade angular;

i - corrente da armadura;

u - tensão aplicada na armadura;

λ - é uma função que modela a característica não linear do torque da carga;

T_e, T_m - são as constantes de tempo elétrica e mecânica, respectivamente.

Vários casos foram simulados para mostrar que o treinamento das redes neurais está bastante relacionado com o grau de não linearidade do sistema físico.

Caso 1 : Neste caso, a constante de tempo mecânica é considerada muito maior que a constante de tempo elétrica de tal forma que as equações do sistema se reduzem a:

$$\frac{d\omega}{dt} = -\omega + \lambda(\omega) + u \quad (56)$$

O torque da carga, neste caso, foi considerado $\lambda(\omega) = \text{sen}(0.1\omega)$ e o sistema foi simulado aplicando-se uma tensão de amplitude aleatória, dentro da faixa de operação de $\pm 50\%$, mantida constante dentro de um período de amostragem equivalente a uma unidade de tempo. A cada amostragem os valores de ω atual e anterior (sistema de ordem igual a 1) são apresentados à entrada da rede neural que é treinada a apresentar, em sua saída, a tensão u aplicada na armadura do motor. Usou-se neste caso uma rede neural com uma única camada intermediária com 50 elementos. As diretivas do capítulo I, para determinação do número de elementos da camada intermediária não foram aplicadas neste caso, visto que, se usou treinamento adaptativo, onde a cada sessão de treinamento da rede são amostrados diferentes valores de entrada e saída do sistema físico. As diretivas do capítulo I se aplicam para o caso em que o treinamento é feito off-line com um conjunto fixo de valores anteriormente amostrados. Assim sendo, o número de elementos na camada intermediária irá definir o grau de precisão na aproximação funcional. Conforme se pode observar pelos resultados obtidos, um mínimo local, equivalente a 0.0012% do valor de operação, foi atingido durante a fase de treinamento (fig. 1) após 7000 apresentações. O uso de um maior número de elementos iria certamente reduzir o valor de erro mínimo atingido. Em aplicações práticas o número de elementos deverá ser escolhido em função do nível de precisão exigido e pelas limitações físicas da rede neural empregada.

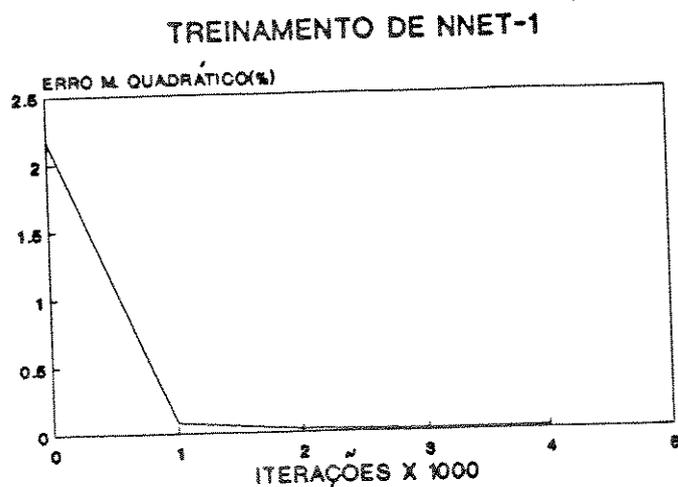


Fig. 1 - Treinamento da rede neural (denominada NNET-1) com a dinâmica inversa de um motor cc, com torque da carga dado por $\lambda(\omega) = \text{sen}(0.1\omega)$.

Após o treinamento o sistema foi simulado novamente e os valores de velocidade atual e anterior, em cada amostragem (sistema de ordem igual a 1) foram apresentados à entrada da rede neural já treinada. A saída da rede foi comparada com a tensão aplicada na armadura do motor e o erro desta comparação está mostrado na figura 2. Na figura 3 apresenta-se a velocidade angular do motor nesta simulação.

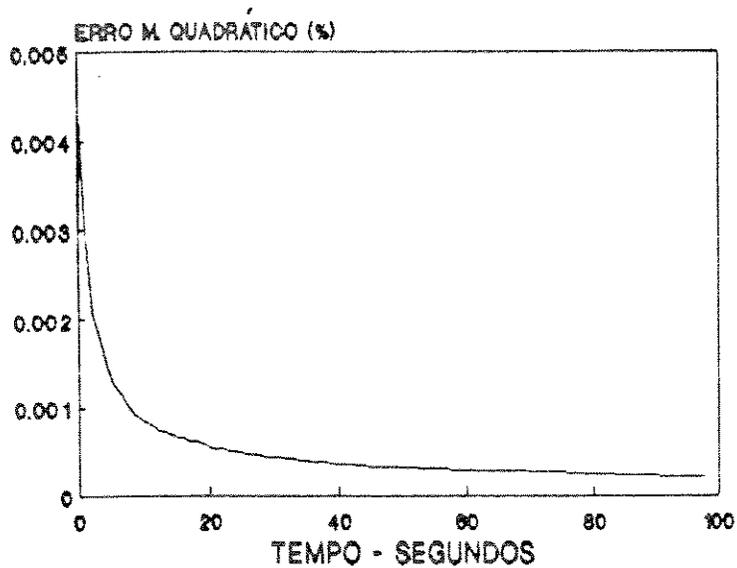


Fig 2 - Desempenho da rede neural treinada com a dinâmica inversa de um motor CC.

Pode-se observar pela figura 2 que a rede neural assimilou a dinâmica inversa do motor de forma bastante precisa, visto que a diferença entre o valor de tensão aplicado ao motor e o valor calculado pela rede neural atingiu valores extremamente pequenos.

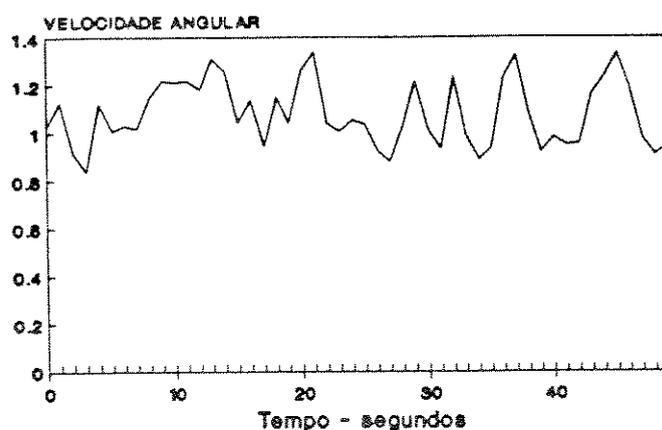


Fig. 3 - Velocidade angular do motor CC excitado com degraus de tensão de amplitude aleatória.

A precisão na identificação da dinâmica inversa é fundamental para o emprego do método de linearização por realimentação apresentado no capítulo IV. Deve-se observar também que, por se tratar de um sistema de tempo contínuo, a rede neural somente poderá ser empregada com o mesmo intervalo de amostragem usado para treinamento da rede. O uso de um intervalo de amostragem diferente daquele usado para treinamento irá certamente invalidar os resultados, ou então inserir imprecisão na determinação da dinâmica inversa.

No caso seguinte, descreve-se a simulação do mesmo motor apresentado anteriormente, considerando-se um grau diferente de não linearidade na carga.

Caso 2: Neste caso a dinâmica inversa do mesmo motor simulado no Caso 1, com uma carga cujo torque é representado por $\lambda(\omega)=\text{sen}(.8\omega)$ foi assimilada por uma rede neural com 100 elementos em sua camada intermediária.

O mesmo procedimento aplicado no caso anterior é usado neste caso. O aumento no número de elementos da camada intermediária é devido ao fato de se ter intensificado o efeito não linear da carga. Desta forma, excitando-se o sistema com o mesmo tipo de tensão na armadura, pode-se atingir o erro médio quadrático de 0.0013%, após 6500 sessões de treinamento. A fig. 4 mostra o decaimento do erro durante a fase de treinamento.

MOTOR CC

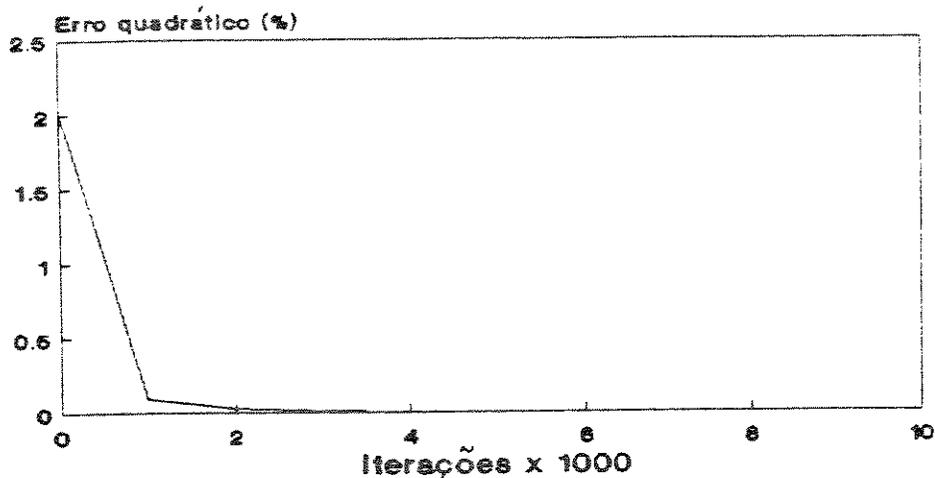


Fig. 4 - Treinamento da rede neural com a dinâmica inversa de um motor CC, com o torque da carga modelado por $\lambda(\omega)=\text{sen}(0.8\omega)$.

Após o treinamento, o sistema foi simulado novamente, apresentando-se à rede, a cada instante, os valores de velocidade em duas amostragens subsequentes. O erro médio quadrático da comparação entre a tensão aplicada e a obtida na saída da rede é apresentada na figura 5. Na figura 6 apresenta-se a velocidade angular do motor durante a simulação.

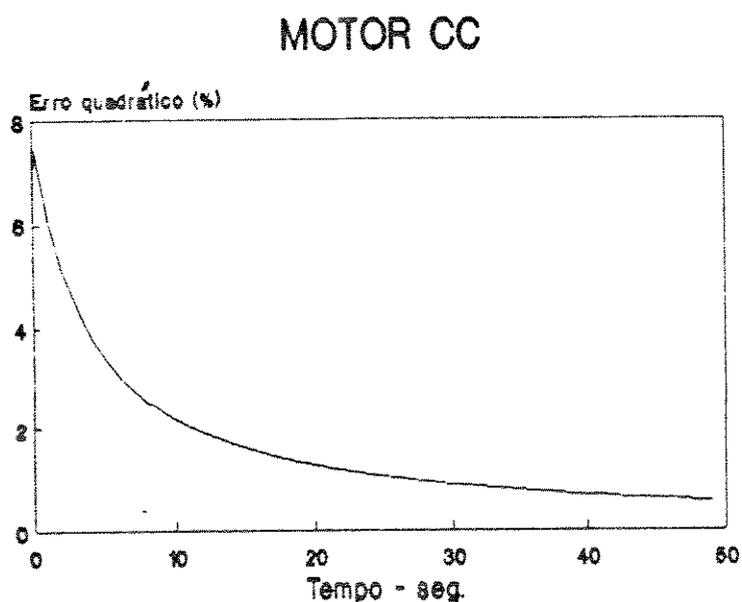


Fig 5 - Desempenho da rede neural treinada com a dinâmica inversa de um motor CC.

MOTOR CC

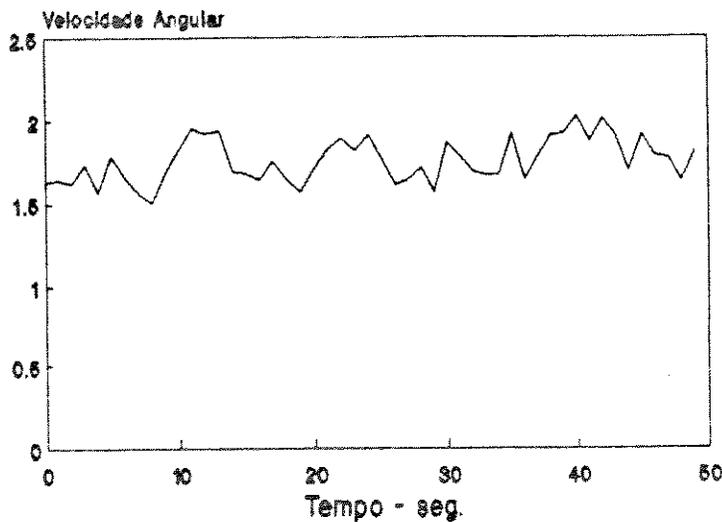


Fig. 6 - Velocidade angular do motor CC excitado com degraus de tensão de amplitude aleatória.

Pode-se observar pelos resultados obtidos que a rede neural assimilou de forma bastante precisa a dinâmica inversa do sistema e que o aumento do nível de não linearidade, exigiu um aumento no número de elementos na camada intermediária, a fim de se atingir a mesma precisão na aproximação funcional. Deve-se salientar também que, por se tratar de um sistema não linear, a dinâmica inversa assim estimada somente deverá ser empregada dentro da faixa de operação usada para treinamento. Embora tais redes possuam excelente capacidade de generalização, deve-se

considerar que não há garantia de operação precisa fora da faixa. Entretanto para alguns sistemas, consegue-se fazer com que a estrutura matemática do sistema seja assimilada pela rede, de forma que se possa usar faixas de operação bastante expandidas em relação a faixa usada na fase de treinamento.

No caso seguinte o motor é modelado com o sistema completo de equações.

Caso 3 - Neste caso, as equações (55a,55b) são usadas para simular o motor cc com a relação $\mu = T_e/T_m = 0.1$. O torque da carga foi representado por $\lambda(\omega) = \text{sen}(.5\omega)$ e se usou uma rede neural com 100 elementos em sua camada intermediária.

Embora se tenha reduzido o nível de não linearidade da carga, manteve-se o mesmo número de elementos na camada intermediária devido ao fato de se tratar de uma equação de ordem mais elevada ($n=2$). Desta forma, excitando-se o sistema com o mesmo tipo de tensão na armadura usado nos casos anteriores pode-se atingir o erro médio quadrático de 0.0011% após 4000 sessões de treinamento. A fig. 7 mostra o decaimento do erro durante a fase de treinamento.

TREINAMENTO DE NNET-1

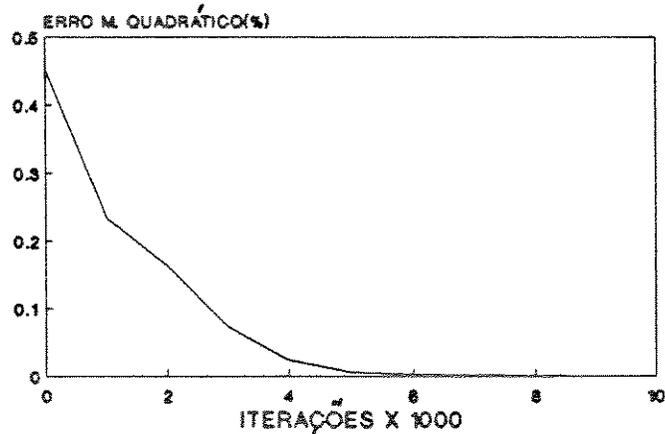


Fig. 7 - Treinamento da rede neural com a dinâmica inversa de um motor CC, com o torque da carga $\lambda(\omega)=\text{sen}(0.5\omega)$ e $\mu=0.1$.

Após o treinamento, o sistema foi simulado novamente, apresentando-se à rede, a cada instante, os valores de velocidade em três amostragens subsequentes. O erro médio quadrático da comparação entre a tensão aplicada e a obtida na saída da rede é apresentado na figura 8. Na figura 9 apresenta-se a velocidade angular do motor durante a simulação.

MOTOR CC

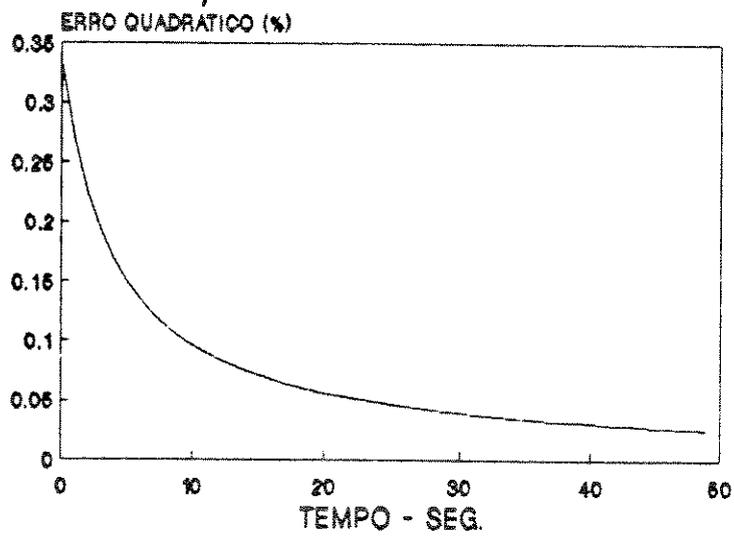


Fig. 8 - Desempenho da rede neural treinada com a dinâmica inversa de um motor CC.

MOTOR CC

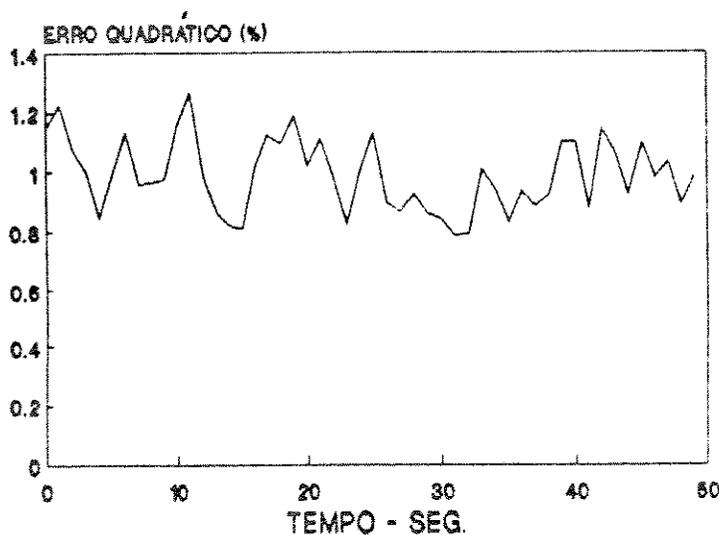


Fig. 9 - Velocidade angular do motor CC excitado com degraus de tensão de amplitude aleatória.

Pode-se observar pelos resultados obtidos que a rede neural assimilou de forma bastante precisa a dinâmica inversa do sistema embora se trate de um sistema de ordem mais elevada que os sistemas treinados nos casos 1 e 2.

A avaliação da dinâmica inversa através de redes neurais para sistemas de tempo discreto é apresentada na próxima seção.

3. Dinâmica Inversa de Sistemas de Tempo Discreto Usando Redes Neurais

Nesta seção, sistemas discretos do tipo Σ_y (eq. II.17) são invertidos usando-se redes neurais. Considerando-se que sistemas do tipo Σ or Σ_n são subclasses do sistema Σ_y , eles podem ser invertidos usando-se o mesmo procedimento desta seção.

Definição 2: A dinâmica inversa do sistema Σ_y é um mapeamento estático que fornece os valores de excitação $u(k)$ que acionam o sistema a partir dos valores de saída $y(k), y(k-1), \dots, y(k-n+1)$, para o valor $y(k+1)$, i.e.:

$$u(k) = \eta^{-1}(y(k+1), y(k), y(k-1), \dots, y(k-n+1)) \quad (57)$$

onde os valores de saída $y(k+1), y(k), y(k-1), \dots, y(k-n+1)$ são medidos nos instantes $k+1, k, k-1, \dots, k-n+1$, e sendo cada um, elementos de um conjunto aberto Y de \mathbb{R}^p . O valor $u(k)$ é a entrada, definida em um conjunto aberto U de \mathbb{R}^m , e η^{-1} é suave.

Proposição 4: Uma rede neural de várias camadas pode ser treinada a aproximar a dinâmica inversa η^{-1} do sistema Σ_y , se um número suficiente de amostras de $y(k+1), y(k), y(k-1), \dots, y(k-n+1)$, e Y e de $u(k) \in U$, for fornecido durante a fase de treinamento.

Demonstração : Sendo η suave, e considerando que redes neurais de várias camadas são aproximadores universais (Cap. I), então a condição suficiente para aproximar a função η^{-1} é apresentada no teorema 1 do capítulo 1, considerando-se que um número suficiente de padrões de treinamento esteja disponível, de forma que a tolerância seja atingida no algoritmo de aproximação. ■

O inteiro n é a ordem do sistema Σ_y , e é por definição, o número suficiente de atrasos de tempo, capaz de definir η de maneira única. Para sistemas do tipo Σ e Σ_n , n é também suficiente, pois o número de atrasos medidos na saída é pelo menos igual à ordem do sistema. Como no caso de tempo contínuo, se o número estimado n for maior que a ordem real do sistema, possivelmente haverá introdução de informação redundante que não irá alterar as propriedades do algoritmo de convergência da rede neural. Portanto, quando n não for disponível, o treinamento deve ser iniciado com um valor pequeno e aumentado quando a tolerância desejada não puder ser atingida.

O exemplo seguinte ilustra a estimação da dinâmica inversa de um sistema de tempo discreto.

Exemplo 2 - Neste exemplo, o método proposto neste trabalho é aplicado para o controle de músculos estimulados eletricamente. Um modelo em tempo discreto, que descreve as propriedades de entrada-saída do músculo foi inicialmente descrito por Chizeck(1989) e analisado em Geng(1989) e Shue(1990), baseado em testes de laboratório com animais vivos. Este modelo foi usado para desenvolver controladores através de linearização por realimentação convencional por El Bially(1990) e através de redes neurais por Teixeira(1991b). O objetivo principal destes testes é o de estabelecer as bases teóricas para aplicação em indivíduos paraplégicos. Estas experiências foram realizadas em conjunto com o Departamento de Engenharia Biomédica e o Departamento de Engenharia de Sistemas da Case Western Reserve University.

Esta aplicação foi motivada pelas experiências bem sucedidas de uso de estímulos elétricos em músculos paralizados para se obter movimentos controlados, como parte de uma

neuro-prótese para restaurar as funções perdidas em indivíduos paralizados, Chizeck(1989) e Shue(1990). Um modelo de seis parâmetros que incorpora as propriedades não lineares de um músculo estimulado eletricamente foi usado, neste exemplo, para gerar os dados de entrada-saída para treinamento das redes neurais. Trata-se de um modelo não linear em tempo discreto, no qual a força do músculo é considerada como o produto de três fatores: ativação, comprimento-tensão e força-velocidade. A dinâmica de ativação é equacionada como um modelo ARMA de segunda ordem. As entradas são pulsos de frequência constante. O desempenho comprimento-força do músculo é aproximado por uma linha reta. Esta aproximação é baseada nas considerações de que a frequência de estímulo seja constante e que as mudanças de comprimento do músculo não sejam muito acentuadas. O termo força-velocidade é aproximado por uma relação linear na qual a inclinação é diferente para velocidade negativa e positiva (isto é, quando o músculo está distendendo ou contraíndo).

Todas estas considerações foram feitas para permitir o uso do modelo matemático definido por Chizeck(1989). Caso o treinamento das redes neurais fosse realizado on-line, o modelo real seria assimilado, sem necessidade de simplificações. Os parâmetros do modelo foram identificados usando-se o método dos mínimos quadrados sequencial não linear com dados obtidos de experiências realizados com músculos de animais vivos (Shue,1990). Como a força e o comprimento do músculo são de difícil medição, foram usados o ângulo de junta e torque. Nesta simulação uma mola simples foi usada para descrever a carga externa. Esta carga relaciona o torque do músculo com a junta resultante e portanto, com o comprimento do músculo.

As equações que representam o modelo são as seguintes, no instante k:

$$y_k = A_k T_{a_k} T_{v_k} \quad (58)$$

$$\phi_k = (1/K_s) Y_k \quad (59)$$

$$A_{k+1} = a_1 A_k + a_2 A_{k-1} + b_0 r_k \quad (60)$$

$$T_{a_{k+1}} = e (\phi_k - \phi^0) \quad (61)$$

$$T_{v_{k+1}} = 1 + h_1 (\phi_k - \phi_{k-1}) u_s (\phi_k - \phi_{k-1}) + h_2 (\phi_{k-1} - \phi_k) u_s (\phi_{k-1} - \phi_k) \quad (62)$$

onde: y_k - torque de saída do músculo;

ϕ_k - ângulo de junta;

ϕ^0 - é um ângulo (não atingível) onde não há torque de saída;

A_k - nível de ativação;

T_{a_k} - termo ângulo-torque;

T_{v_k} - termo torque-velocidade;

K_s - constante da mola representando a carga do músculo;

$u_s ()$ - função degrau unitário;

r_k - estímulo aplicado ao músculo;

$e, a_1, a_2, b_0, h_1, h_2$ - são parâmetros do modelo.

Uma sequência de números aleatórios foi usada como sinal de entrada, escalados apropriadamente para corresponder a largura de pulso usado na prática.

Neste exemplo serão apresentados os resultados obtidos com na determinação da dinâmica inversa do modelo e no capítulo IV será apresentada a simulação da linearização por realimentação.

O sistema foi simulado e os valores de saída de y_{k+1} , y_k , y_{k-1} foram usados como entrada da rede neural. O estímulo r_k foi usado como saída desejada da rede neural com 100 elementos na camada intermediária. As figuras 10 e 11 apresentam o decaimento

do erro durante a fase de treinamento e o desempenho da rede na fase de teste, respectivamente.

Modelo de um músculo

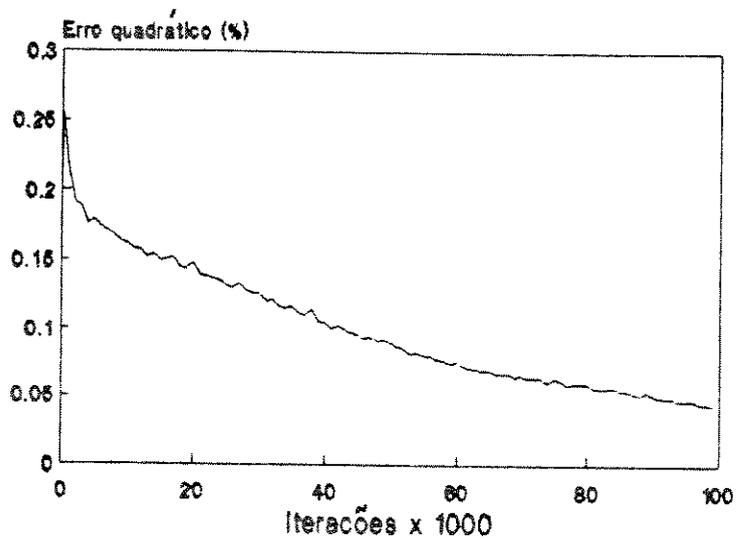


Fig. 10 - Decaimento do erro quadrático no treinamento da dinâmica inversa de um músculo estimulado eletricamente.

Modelo de um músculo

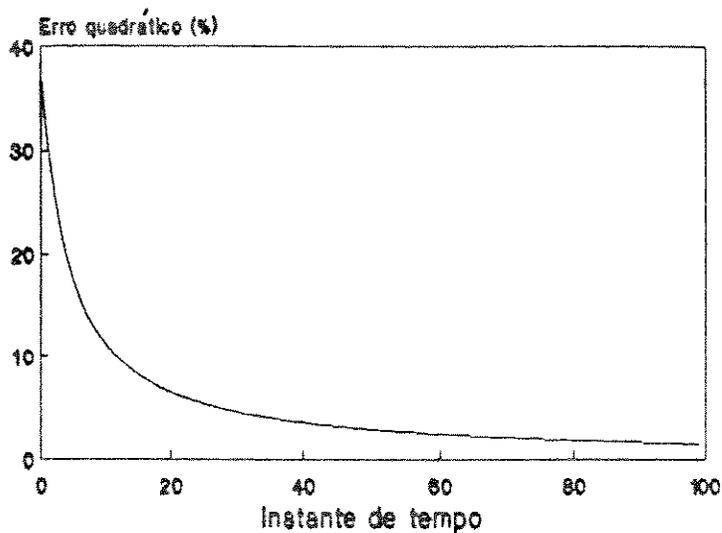


Fig. 11 - Desempenho da rede neural para o estimativa da dinâmica inversa de um músculo estimulado eletricamente.

O resultados apresentados neste exemplo serão usados no próximo capítulo onde o mesmo modelo é aplicado em um esquema de linearização por realimentação. Embora tenha sido aplicado um modelo matemático para o treinamento da rede, o método sugere ser bastante promissor para aplicações on-line, desde que esteja disponível um hardware de rede neural adequado para implementação direta.

4. Conclusão

Neste capítulo, foi proposto um método para se determinar a dinâmica inversa de sistemas não lineares. O método tanto pode ser aplicado para sistemas de tempo discreto como contínuo. Para o caso de tempo contínuo, foi apresentado um método alternativo para se evitar medições de derivadas da saída. O desempenho das redes foi testado através de exemplos e os resultados das simulações mostraram que a generalização desejada foi atingida.

CAPITULO IV

Linearização Por Realimentação Usando Redes Neurais

Este capítulo apresenta um método para se controlar sistemas não lineares através de um esquema adaptativo com uso de redes neurais. Para uma classe restrita de sistemas, são apresentadas condições necessárias para se ensinar o mapeamento de entrada-saída a uma rede neural de várias camadas, treinada pelo algoritmo de propagação retroativa (regra delta generalizada).

O projeto de sistemas robustos de controle para sistemas dinâmicos lineares tem se constituído uma área de pesquisa bastante ativa. Para sistemas dinâmicos não lineares, poucos métodos são atualmente disponíveis. Em termos específicos, um dos maiores problemas é o tratamento de sistemas não lineares com estrutura e parâmetros desconhecidos. Este capítulo apresenta uma técnica para controle por modelo de referência, aplicável a uma classe específica de sistemas não lineares, através do uso de redes neurais.

Pelo método proposto, duas redes neurais são treinadas em dois diferentes estágios. No primeiro estágio, o sistema não linear é excitado várias vezes para se ensinar a sua dinâmica inversa a uma rede neural (veja capítulo III). No segundo estágio, o sistema é novamente excitado várias vezes para se treinar a segunda rede neural que irá controlá-lo na forma desejada. Depois dos dois primeiros estágios de treinamento, o sistema é operado com a segunda rede neural, na realimentação, e seus pesos são adaptativamente ajustados para acomodar possíveis variações de parâmetros no sistema não linear.

Para um classe específica de sistemas não lineares, são estabelecidos esquemas de treinamento das redes neurais, e são também apresentados resultados de simulação preliminares onde sistemas não lineares são linearizados por realimentação. Embora qualquer modelo de referência possa ser usado, este trabalho focaliza a discussão na aplicação de um modelo linear de referência, pois desta forma se estabelecem condições para se linearizar o sistema. No capítulo II deste trabalho, apresenta-se uma discussão de linearização por realimentação para sistemas de tempo contínuo ou discreto.

O método de linearização apresentado neste capítulo, pode ser aplicado para sistemas de tempo contínuo do tipo ψ (eq. II.1) ou ψ_N (eq. II.8), ou sistemas de tempo discreto do tipo Σ (eq. II.12) ou Σ_N (eq. II.16). Inicialmente, o método é apresentado para sistemas de tempo discreto, onde condições necessárias são estabelecidas e os resultados de simulações são apresentados.

1. Linearização por realimentação em sistemas não lineares de tempo discreto através do uso de redes neurais

Um dos principais problemas referentes ao controle de sistemas não lineares é a identificação do sistema. Loparo & Teixeira (1990) apresentam um método de linearização por realimentação em sistemas não lineares não identificados, usando redes neurais, e baseando-se na identificação da dinâmica inversa do sistema. Narendra & Parthasarathy (1990), apresentam quatro modelos para identificação de sistemas discretos não lineares, também através do uso de redes neurais. Um esquema adaptativo, que aplica um modelo de referência como controle, também é

apresentado no trabalho de Narendra. Chen (1990) apresenta um método de controle auto-ajustável para sistemas não lineares, usando redes neurais. No trabalho de Chen, sistemas de tempo discreto do tipo Σ são identificados usando-se redes neurais de várias camadas e em seguida um processo de linearização é aplicado.

Neste capítulo, um método diferente é proposto: em vez de se identificar a dinâmica direta do sistema, a dinâmica inversa é identificada por uma rede neural que é usada para treinar uma outra rede neural que irá realmente linearizar o sistema. Este método pode ser aplicado para uma classe mais geral de sistemas, tanto de tempo discreto como contínuo. Na seção seguinte sistemas de tempo discreto são considerados.

1.1 Linearização de Entrada-Saída por Realimentação em Sistemas de Tempo Discreto

Nesta seção, são tratados sistemas do tipo Σ_y (eq. II.17), que podem ser representados em termos de uma equação de entrada-saída a diferenças. O esquema básico de controle, consiste em se aplicar uma realimentação ao sistema Σ_y para forçá-lo a seguir o desempenho de um sistema linear de referência do tipo Γ , definido pela equação II.18. Este sistema pode ser escrito na forma de uma equação de entrada-saída, como definido na equação II.19.

Neste capítulo, em vez de se aplicar o processo analítico apresentado no capítulo II, são usadas redes neurais para aproximar as funções de linearização. O processo de linearização pode tanto ser realizado adaptativamente, onde os pesos das conexões da rede neural são ajustados durante a operação

normal do sistema físico, como de uma forma não adaptativa, onde os pesos são ajustados off-line. Em ambos os casos, pode-se distinguir três fases, embora as fases 1 e 2 sejam idênticas para os casos adaptativo e não adaptativo.

Por definição, a função inversa (eq. III.57) determina a entrada $u(k)$ que transfere o sistema Σ_y dos valores prévios de saída $y(k), y(k-1), \dots, y(k-n+1)$ para o valor $y(k+1)$. Considera-se que a função η^{-1} seja aproximada através de uma rede neural de várias camadas sob a tolerância τ , quando treinada com um número suficiente de amostras do espaço de definição da saída Y^{n+1} (veja capítulo 1 e 3).

Definição 1 - A função de linearização por realimentação γ é um mapeamento do produto espacial $Y^{n+1} \times V$ em U tal que,

$$u(k) = \gamma(v(k), y(k), y(k-1), \dots, y(k-n+1)) \quad (01)$$

A função γ é considerada um mapeamento suave relacionando os últimos n valores de saída $y(l)$, $l=k-n+1, \dots, k$, e a entrada $v(k)$ ao sinal atuante $u(k)$, onde Y, U, V são subconjuntos abertos em \mathbb{R}^p e \mathbb{R}^n respectivamente. ■

O objetivo é determinar $u(k)$ tal que, quando aplicado ao sistema Σ_y , faça com que a relação de entrada-saída $\{v(k), y(k+1)\}$ siga a relação de entrada-saída $\{v(k), y_L(k+1)\}$, do sistema linear Γ_y , isto é, $y(k+1) = y_L(k+1)$.

Sendo a função de realimentação γ uma função suave, pode-se aproximá-la por uma rede neural com pelo menos uma camada intermediária, conforme mostrado no capítulo I, considerando-se que um número suficiente de amostras dos espaços de definição

γ^{n+1} x V e U estejam disponíveis para treinamento da rede.

Proposição 1: O sinal atuante,

$$u(k) = \eta^{-1}(y_L(k+1), y(k), y(k-1), \dots, y(k-n+1)) \quad (02)$$

aciona o sistema não linear Σ_y dos valores prévios de saída $y(k), y(k-1), \dots, y(k-n+1)$, para o valor $y(k+1) = y_L(k+1)$, onde $y_L(k+1)$ é a saída do sistema linear Γ_y acionado pela entrada $v(k)$, a partir dos valores prévios de saída $y_L(k), y_L(k-1), \dots, y_L(k-n+1)$.

Demonstração: A saída $y(k+1)$ do sistema não linear Σ_y , para $y(k)=y_L(k), y(k-1)=y_L(k-1), \dots, y(k-n+1)=y_L(k-n+1)$ é dada por,

$$y(k+1) = \eta(y_L(k), y_L(k-1), \dots, y_L(k-n+1), u(k)). \quad (03)$$

Como se deseja $y(k+1)=y_L(k+1)$, a função inversa se torna,

$$u(k) = \eta^{-1}(y_L(k+1), y_L(k), y_L(k-1), \dots, y_L(k-n+1)) \quad (04)$$

portanto torna-se suficiente aplicar $u(k)$, calculado por (04), ao sistema Σ_y para se ter $y(k+1) = y_L(k+1)$, sendo necessário que η seja uma função suave um para um.

Proposição 2: Uma rede neural com pelo menos uma camada intermediária pode ser treinada para aproximar a função γ , que lineariza por realimentação o sistema Σ_y , considerando-se que os seguintes dados estejam disponíveis para treinamento:

Saída: $u(k) = \eta^{-1}(y_L(k+1), y(k), y(k-1), \dots, y(k-n+1))$, isto é, a dinâmica inversa do sistema Σ_y acionada pela saída $y_L(k+1)$ do

sistema linear Γ_y , usado como modelo de referência e os valores prévios de saída $y(k), y(k-1), \dots, y(k-n+1)$, provenientes do sistema Σ_y .

Entrada: o sinal $v(k)$, que corresponde à entrada do sistema linear Γ_y , e os valores prévios de saída $y(k), y(k-1), \dots, y(k-n+1)$, provenientes do sistema Σ_y .

Demonstração: as condições necessárias são providas pela proposição 1, onde $u(k)$ se constitui na entrada capaz de linearizar o sistema Σ_y . Entretanto para se ter Σ_y linearizado por realimentação torna-se também necessário que a dinâmica zero de Σ_y seja assintoticamente estável, Sastry & Isidori (1989). Considerando-se serem as redes neurais de várias camadas aproximadores universais (capítulo I), a condição de suficiência estará satisfeita, tomando-se um número de amostras que seja representativo dos subespaços de definição, isto é:

$$\{y(k), y(k-1), \dots, y(k-n+1)\} \in Y^n \text{ e } u(k) \in U. \blacksquare$$

As proposições 1 e 2 apresentam as bases para linearização por realimentação de uma classe de sistemas não lineares através do uso de redes neurais de várias camadas. Este processo pode ser realizado em três fases distintas para o caso não adaptativo.

Caso Não-Adaptativo: A primeira fase consiste em se ensinar uma rede neural NNET-1 (fig. 1), com a dinâmica inversa do sistema não linear Σ_y . Várias amostras de entrada-saída do sistema Σ_y são tomadas e NNET-1 é treinada usando-se as diretivas do capítulo I. A entrada para a rede NNET-1 consiste das últimas saídas do sistema Σ_y , como mostrado no capítulo III. A entrada para NNET-2,

entretanto, consiste dos r últimos valores de saída de Σ_y e da referência $v(k)$. O inteiro r é definido como:

$$r = \max(r_1, r_2, \dots, r_m) \quad (05)$$

onde r_1, r_2, \dots, r_m correspondem ao vetor de grau relativo do sistema Σ_y , e m é a dimensão da entrada.

A amostragem é realizada excitando-se o sistema Σ_y com entradas aleatórias em um período que seja suficiente para que a rede neural possa captar a estrutura da dinâmica inversa do sistema. Se as amostras de saída fornecerem uma representação suficiente do produto espacial $Y^{n+1} \times U$, então a rede neural também fornecerá uma representação razoável da dinâmica inversa do sistema. No capítulo I são apresentadas as diretivas para se efetuar a amostragem e para se dimensionar a rede neural.

Basicamente neste caso, o treinamento de NNET-1 com a dinâmica inversa, é realizado como ilustrado na fig. 1. O sistema Σ_y é excitado com uma entrada aleatória, até que se atinja convergência aceitável da rede (veja capítulo III).

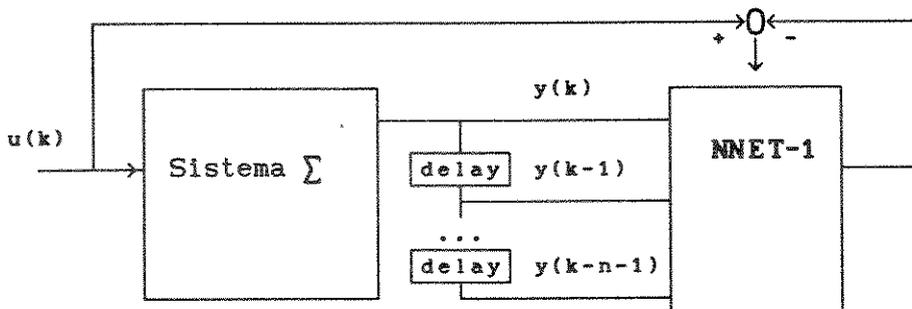


Fig. 1 - Treinamento de NNET-1 com a dinâmica inversa.

Na segunda fase, executa-se o treinamento de NNET-2

(também uma rede neural de várias camadas) usando-se o esquema ilustrado na fig. 2. O valor u_d é a saída desejada para NNET-2, isto é, o valor de excitação que faz com a saída y do sistema não linear siga o modelo y_L .

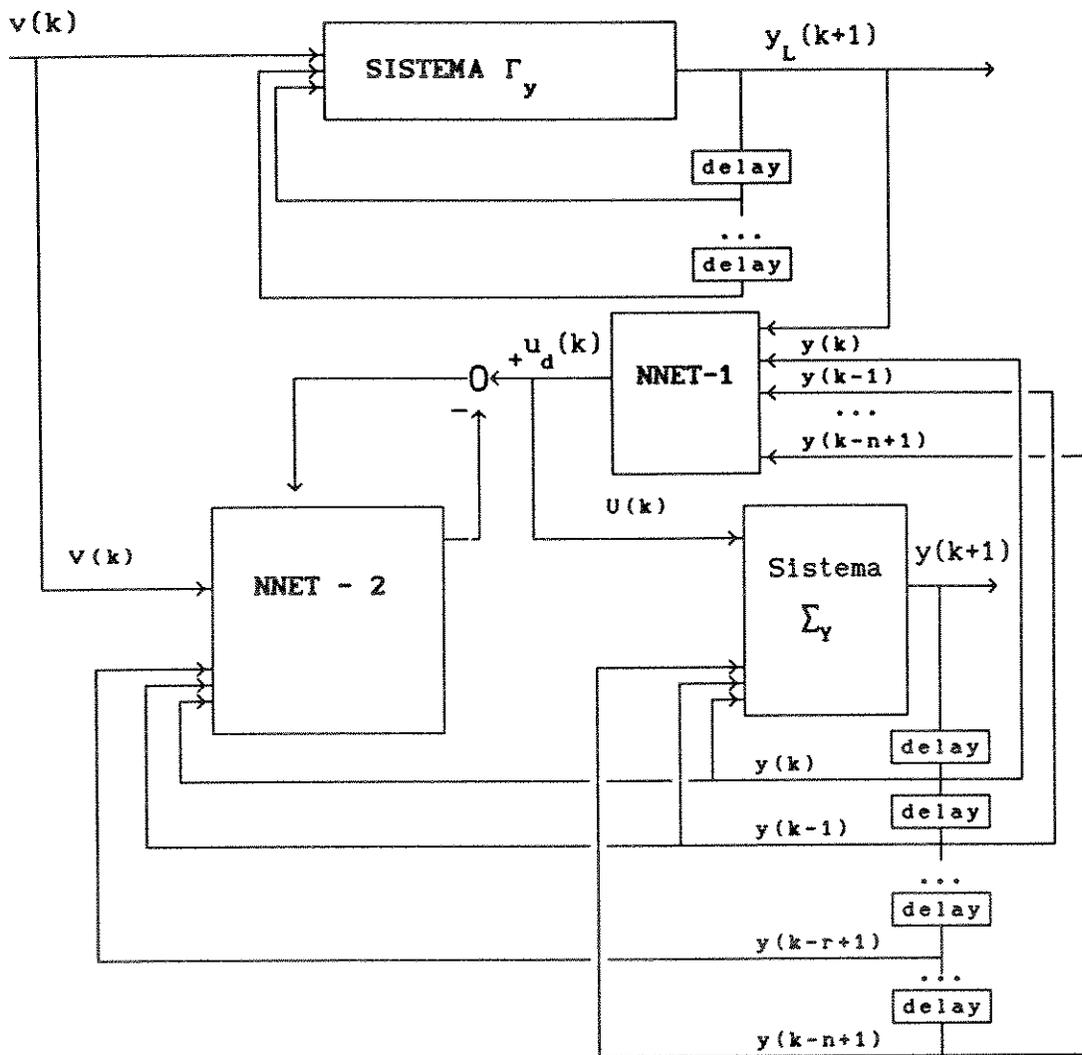


Fig. 2 - Treinamento de NNET-2, a rede neural que lineariza por realimentação.

Nesta fase, a rede neural NNET-1, previamente treinada, é usada para gerar a saída desejada de NNET-2 (veja fig.2). O sistema linear Γ_y é simulado com valores aleatórios de entrada $v(k)$ e sua saída $y_L(k+1)$ é alimentada em NNET-1. O sistema Σ_y é também acionado e suas saídas $y(k), y(k-1), \dots, y(k-n-1)$ são introduzidas em NNET-1. Considerando-se que NNET-1 aproxima a dinâmica inversa do sistema Σ_y , então,

$$u_d(k) = \eta^{-1}(y_L(k+1), y(k), y(k-1), \dots, y(k-n+1)) \quad (06)$$

Isto significa que $u_d(k)$ é o valor de entrada que faz com que a saída do sistema Σ_y aproxime o valor $y_L(k+1)$, a partir dos valores $y(k), y(k-1), \dots, y(k-n+1)$.

O valor de $u_d(k)$ é usado como valor desejado, para treinar NNET-2, a rede neural de linearização por realimentação. As entradas para NNET-2 são os valores de saída do sistema Σ_y , especificamente, $y(k), y(k-1), \dots, y(k-n+1)$ e a excitação $v(k)$. NNET-2 calcula o valor de $u(k)$ que é usado para acionar o sistema Σ_y . A condição necessária para aplicação deste método é que a dinâmica nula do sistema Σ_y seja estável. Além do mais, o sistema deverá ser simulado por longos períodos e portanto a amplitude máxima dos valores de saída deverá ser limitada.

Depois que o treinamento de NNET-2 esteja completo, pode-se iniciar a fase 3, que se constitui na operação normal do sistema. A figura 3 ilustra esta fase, mostrando como a rede NNET-2 é usada para determinar o sinal atuante do sistema Σ_y .

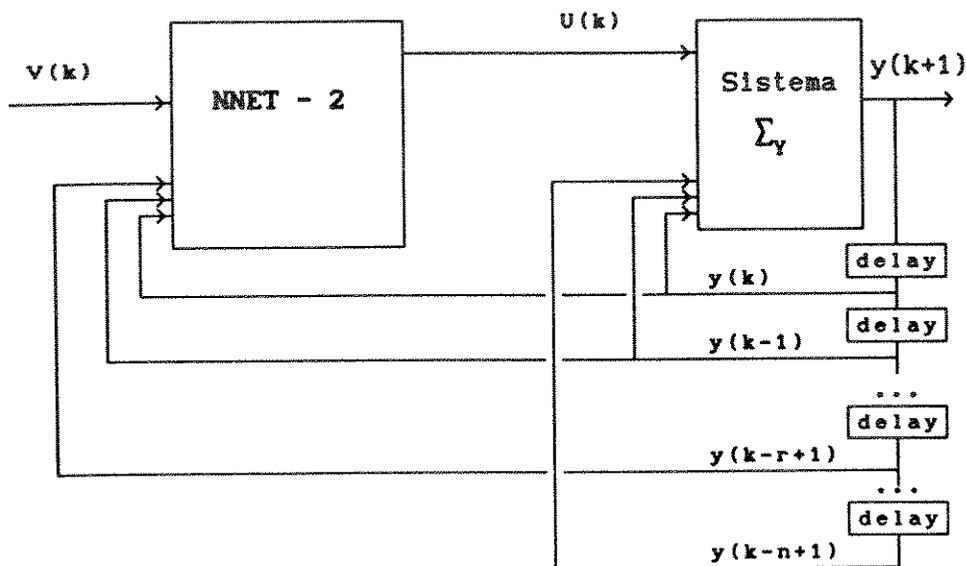


Fig.3 - NNET-2 aplicada para linearizar por realimentação o sistema Σ_y .

Observe que a rede neural da dinâmica inversa, NNET-1, não é usada nesta fase, pois esta rede somente é útil, neste tipo de operação, para ajustar os pesos das conexões de NNET-2. Isto requer que os parâmetros do sistema sejam fixos e que o sistema seja operado no espaço de definição usado para treinar NNET-1 e NNET-2.

Caso Adaptativo: Neste modo de operação, as redes neurais são inicialmente treinadas off-line, como no caso não-adaptativo. A instalação é então colocada em operação com todas as redes neurais conectadas. Seus pesos são sistematicamente ajustados para acompanhar as variações do modelo, ou de parâmetros do modelo. Entretanto, este esquema somente poderá ser aplicado para sistemas com variações lentas de parâmetros. Essencialmente, as variações de parâmetros devem ocorrer em uma escala de tempo que seja maior

que o período requerido para treinamento das redes neurais. Basicamente, o processo consiste de três fases:

Fase 1: Estimação off-line da dinâmica inversa do sistema, i.e., ajuste inicial de NNET-1, como no caso não-adaptativo.

Fase 2: Estimação do mapeamento de linearização, isto é, ajuste inicial de NNET-2, como no caso não-adaptativo.

Fase 3: Operação normal e ajuste adaptativo de NNET-1 e NNET-2.

Na terceira fase, depois de p intervalos de tempo, uma ou mais iterações do algoritmo de propagação retroativa são aplicadas às redes NNET-1 e NNET-2. O inteiro p deve ser escolhido com base em conhecimento prévio do desempenho do sistema. Isto somente é possível se o intervalo de amostragem for suficientemente longo para permitir que um ou mais passos de propagação retroativa sejam aplicados. Torna-se também necessário que as variações de parâmetros ocorram de forma a serem captadas pela rede neural em poucas iterações. Contudo, se o hardware disponível de rede neural for suficientemente rápido, poder-se-á atingir convergência no tempo disponível entre duas amostragens.

Indiscutivelmente, nos esquemas de controle que aplicam a dinâmica inversa no ramo direto, há a possibilidade de ocorrência de instabilidade como resultado da presença de altas frequências. Estas frequências são incluídas devido ao fato de não ser possível um cancelamento exato das não-linearidades. Assim sendo, espera-se que o esquema apresentado seja mais robusto, pois a função dinâmica inversa não é diretamente usada do ramo de controle. Estas questões referentes a robustez do sistema deverão ser investigadas em um outro trabalho. A figura 3 ilustra a

aplicação do esquema adaptativo. Note que durante o treinamento, a primeira entrada de NNET-1 deve ser transferida de $y_L(k+1)$ para $y(k+1)$, para atualizar a dinâmica inversa de Σ_y .

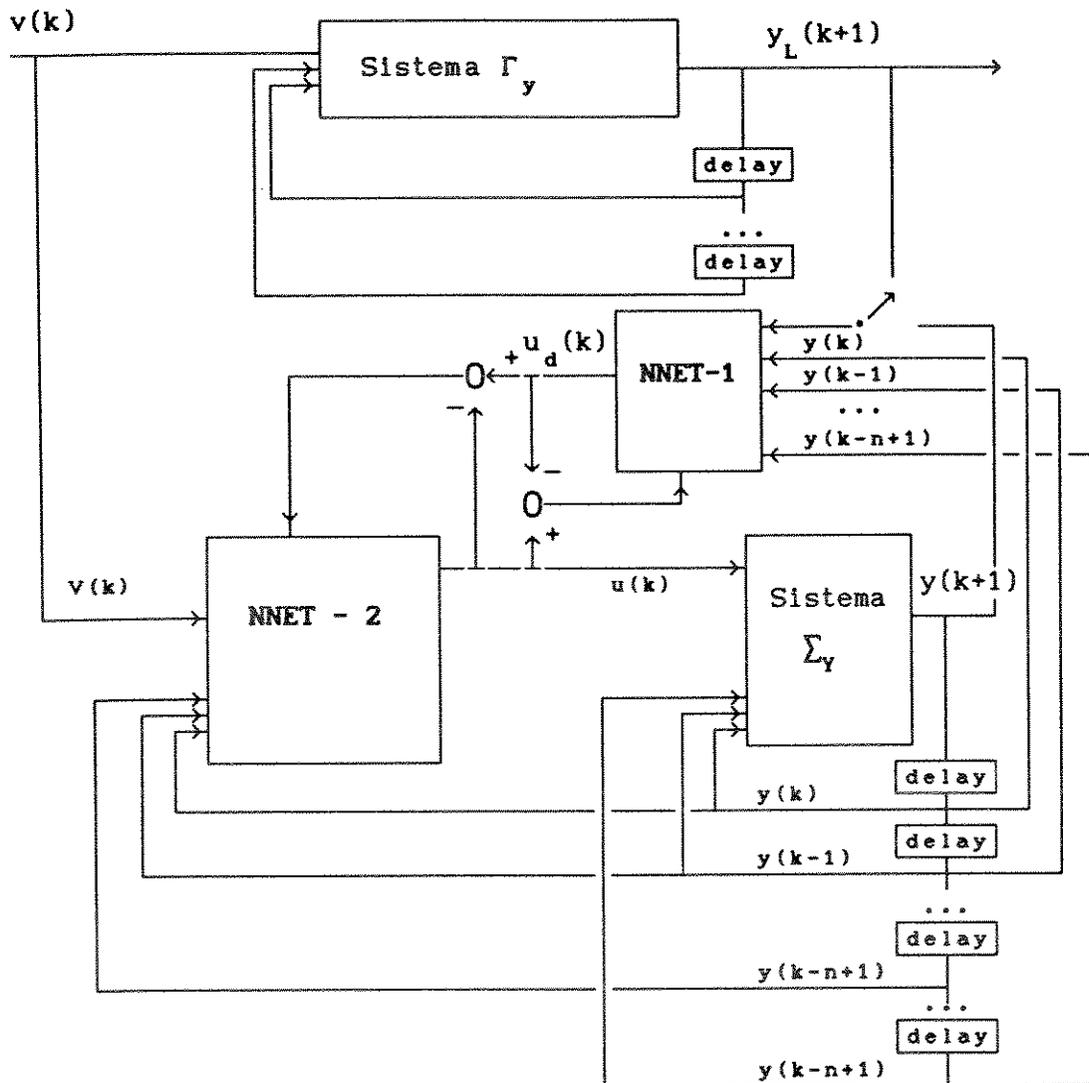


Fig.4 - Treinamento adaptativo de NNET-1 e NNET-2.

O método de linearização por realimentação através de

redes neurais, também pode ser estendido para se obter a transformação de coordenadas capaz de linearizar o sistema no espaço de estado. A seção seguinte introduz esse método.

1.2 Linearização por realimentação no espaço de estados para sistemas de tempo discreto.

Basicamente, a diferença entre linearização por realimentação de entrada-saída e linearização por realimentação no espaço de estado é que no último, o sistema se torna linear do ponto de vista da entrada para o estado do sistema. Em alguns casos é possível se obter um desempenho linear do ponto de vista entrada-estado-saída (veja capítulo II). No caso de linearização de entrada-saída por realimentação, os valores de estado não são importantes, e o desempenho do sistema em termos da resposta entrada-estado, não se torna necessariamente linear.

Se o estado do sistema for mensurável, o procedimento básico de controle a ser aplicado neste caso, será aplicar uma realimentação de estado e uma mudança de coordenadas ao sistema Σ_N para fazê-lo seguir o desempenho do sistema linear Γ (eq. II.18). O método é similar à linearização de entrada-saída, exceto que uma outra fase é adicionada para treinar uma terceira rede neural, NNET-3 (veja fig. 7), para aproximar a transformação de coordenadas. As fases 1 e 2 são idênticas às fases do primeiro método. NNET-3 aproxima a função $\phi: x \rightarrow z$ (veja capítulo II), correspondente a transformação de coordenadas que calcula o estado do sistema linear a partir do estado do sistema não linear. A seguinte proposição estabelece a base teórica para este método.

Proposição 3: Uma rede neural de várias camadas com pelo menos uma camada intermediária pode ser treinada para aproximar a função γ

que lineariza por realimentação o sistema Σ_N , considerando-se que os seguintes dados sejam usados para treinamento da rede.

Saída: $u(k) = \zeta^{-1}(y_L(k+1), y(k), y(k-1), \dots, y(k-n+1))$, isto é, a dinâmica inversa do sistema Σ_N alimentada com a saída $y_L(k+1)$ do sistema linear Γ , usado como modelo, e as saídas prévias $y(k), y(k-1), \dots, y(k-n+1)$ do sistema Σ_N .

Entrada: a entrada $v(k)$ do sistema linear Γ , e o estado prévio $x(k)$ do sistema Σ_N .

Demonstração: A proposição 2 estabelece conclusões similares, exceto que γ foi definida em termos de $v(k)$ e dos valores prévios de $y(k), y(k-1), \dots, y(k-n+1)$. Considere que o sistema seja observável e que a função $h(x)$ seja suave. Assim sendo, dado o estado do sistema, a saída poderá ser calculada pela proposição 2, e uma rede neural poderá ser treinada para aproximar a função de linearização γ . Neste caso, valores de estado em vez de valores de saída são usados e as mesmas condições de suficiência da proposição 2 podem ser aplicadas. ■

Embora uma realimentação de estado esteja sendo usada para treinar a rede neural, o sistema não se tornará necessariamente linear em termos da resposta entrada-estado. Somente a função de linearização, $\gamma(x, v)$, não será capaz de linearizar o sistema do ponto de vista da resposta entrada-estado. Haverá necessidade de uma mudança de coordenadas.

Pode-se resumir nos seguintes passos, os procedimentos para linearização por realimentação no espaço de estado, para o caso não adaptativo:

Fase 1: Treinamento da dinâmica inversa da rede neural NNET-1

Esta fase é realizada da mesma forma que no caso de linearização de entrada-saída por realimentação (vide fig. 1).

Fase 2: Treinamento da rede de linearização de entrada-saída por realimentação- NNET-2 .

Neste caso, a rede NNET-2 transforma o sistema Σ_N em um sistema linear em termos da resposta entrada-saída. A figura 4 mostra as conexões necessárias para se efetuar o treinamento da rede.

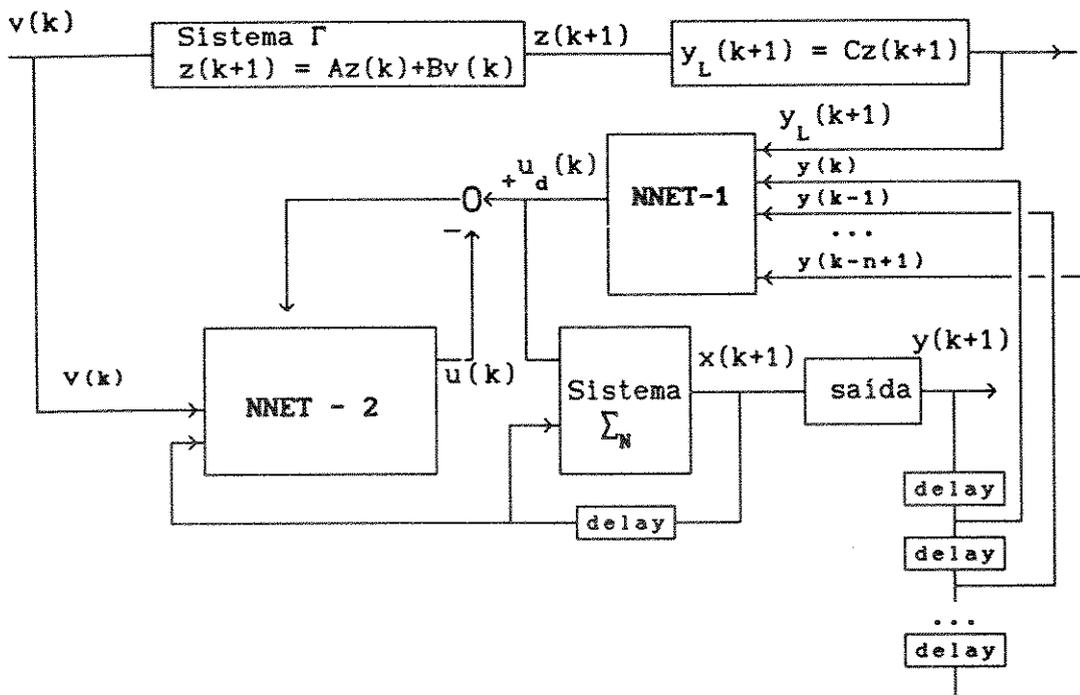


Fig.5 - Treinamento de NNET-2, baseado em valores de estado.

Embora o objetivo seja obter linearização entrada-estado, nada impede que ao invés de se realimentar NNET-2 com o estado do sistema, se alimente com os valores necessários de saída, como no caso de linearização por realimentação de entrada-saída. Na maioria dos casos práticos isto é mandatório visto que não se tem acesso ao estado do sistema.

Fase 3: Treinamento da transformação de coordenadas à rede NNET-3

Depois da fase 2, o sistema é operado em malha fechada com a rede NNET-2. O estado de Σ_N é alimentado em NNET-3 e sua saída é comparada com o estado z do sistema linear Γ como ilustrado na fig. 6. Depois de completado o treinamento, a rede NNET-3 terá sido treinada com uma aproximação da transformação de coordenadas apropriada ϕ , que é necessária para o processo de linearização por realimentação no espaço de estado (vide capítulo II).

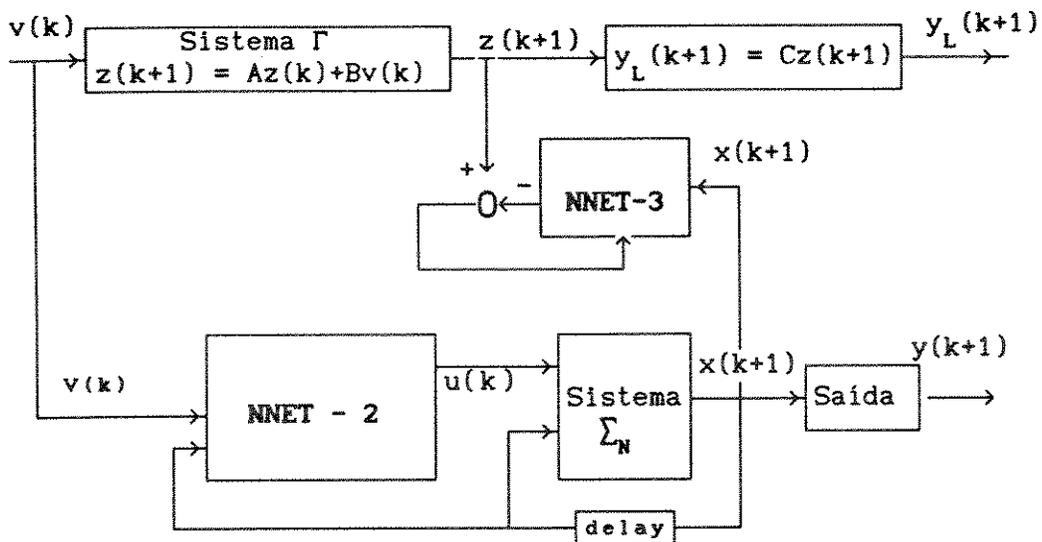


Fig.6 - Treinamento da rede neural para transformação coordenadas.

Para uma classe restrita de sistemas, a transformação de coordenadas poderá ser obtida, a partir de valores de saída do sistema não linear. Isto se aplica, para o caso em que não se tenha acesso físico ao estado do sistema. Desta forma, a rede NNET-3 seria alimentada com valores de saída do sistema não linear e seria, portanto, treinada a apresentar em sua saída os valores do estado do sistema linear. Evidentemente, este processo somente é aplicável para sistemas observáveis. A abrangência da aplicação deste processo carece de uma investigação mais intensa e não será tratada neste trabalho. Entretanto, para o caso de sistemas, sem acesso físico ao estado, o método se torna uma alternativa viável.

Fase 4: Operação normal do sistema com NNET-2 e NNET-3.

Nesta fase as redes já treinadas são conectadas para operação normal, como mostrado na fig. 6. Note que NNET-1 não é usada e o sistema se comporta de forma linear da entrada para o estado (v para z).

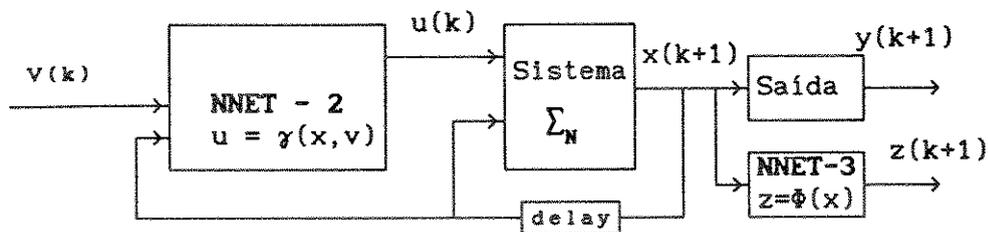


Fig. 7 - Operação Normal: linearização por realimentação relacionando entrada-estado-saída.

Se a alternativa de alimentação das redes a partir da saída, em vez de estado for escolhida o esquema da figura 7 deverá ser modificado.

Desta forma, considerando-se que de v para z o sistema se comporta de forma linear, qualquer controle tradicional para sistemas lineares poderá ser aplicado.■

No esquema adaptativo (fig. 8), há três redes neurais presentes, e depois de p intervalos de tempo, uma ou mais iterações de propagação retroativa são aplicadas para ajustar as redes NNET-1, NNET-2 e NNET-3. O inteiro p é escolhido usando-se o conhecimento disponível do comportamento do sistema. Certamente, isso somente é possível se os intervalos de tempo entre as amostragens forem suficientemente longos para permitir que um ou mais passos de propagação retroativa sejam aplicados. Entretanto, o hardware de rede neural disponível poderá ser suficientemente rápido para permitir que uma ou mais sessões de treinamento sejam realizadas no tempo disponível entre duas amostragens. Assim sendo, da mesma forma que no caso de linearização de entrada-saída, torna-se necessário que as variações de parâmetros ocorram em um período suficientemente longo para permitir que poucas iterações de propagação retroativa sejam suficientes.

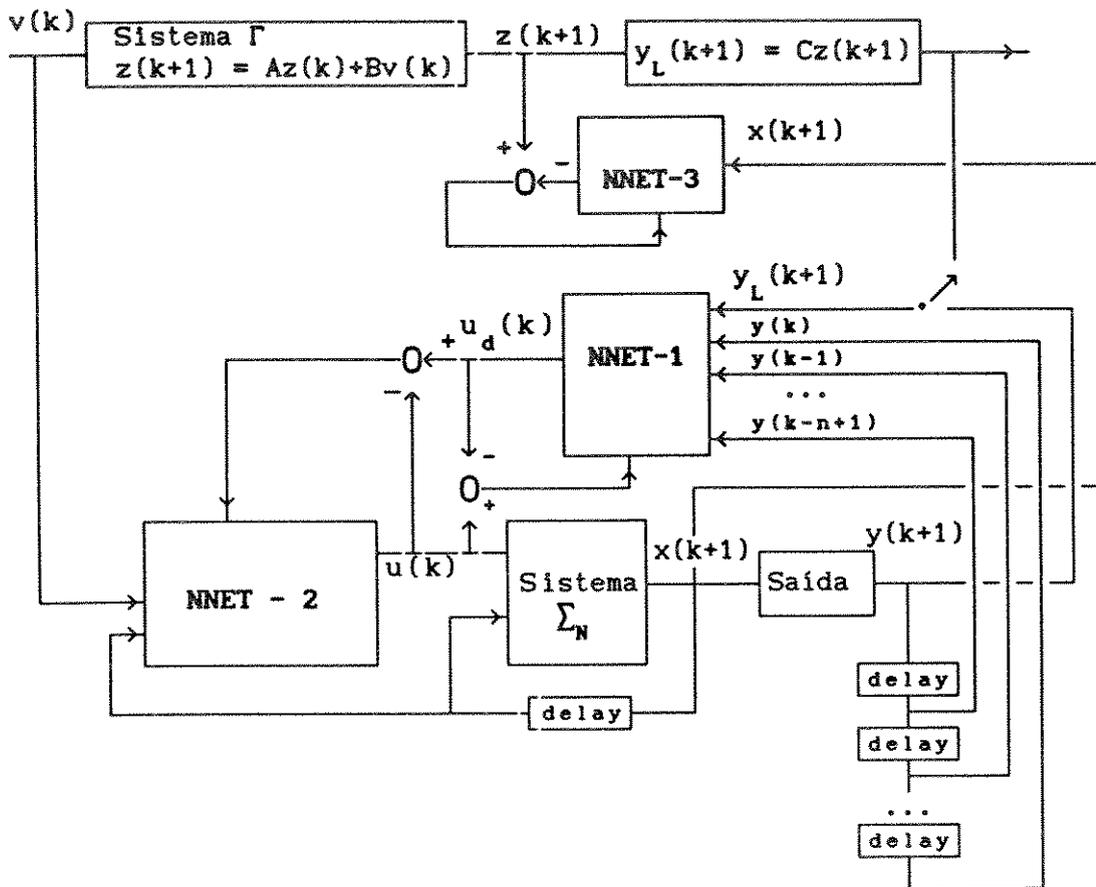


Fig.8 - Treinamento adaptativo de NNET-1, NNET-2 e NNET-3.

Note que para se aplicar algumas iterações de propagação retroativa à rede NNET-1, a primeira entrada de NNET-1 precisa ser transferida da saída do sistema linear para a saída do sistema não linear.

O exemplo seguinte ilustra a aplicação de redes neurais para sistemas de tempo discreto.

Exemplo 1 - O sistema que representa um músculo eletricamente estimulado, apresentado no capítulo III (exemplo 2) e cuja dinâmica inversa foi identificada, é linearizado por realimentação, neste exemplo. O esquema da figura 2 foi aplicado, usando o seguinte sistema como modelo linear, após o treinamento da rede NNET-1:

$$y_L(k+1) = 0.6y_L(k) - 0.05y_L(k-1) + 0.025v_k \quad (06)$$

onde $y_L \in \mathbb{R}$ é a variável de saída e $v \in \mathbb{R}$ é o controle externo aplicado ao sistema.

O sistema foi excitado usando-se valores aleatórios dentro de uma faixa de operação de $\pm 20\%$, em torno do ponto de operação $v = 100v$. Na figura 9 apresenta-se o decaimento do erro médio quadrático, na fase de treinamento da rede.

Modelo de um músculo

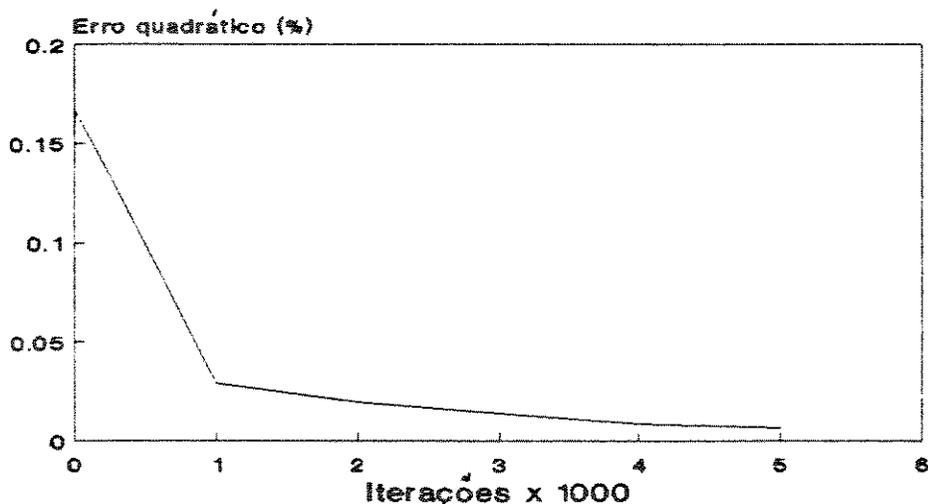


Fig. 9 - Decaimento do erro na fase de treinamento da rede NNET-2 para linearização do modelo não-linear de um músculo estimulado eletricamente.

Em seguida o sistema foi simulado novamente segundo o esquema da figura 3, onde a rede NNET-2, já treinada, é aplicada para linearizar o sistema por realimentação. Na fig. 10 apresenta-se o erro médio quadrático da comparação entre a saída do sistema não linear e a saída do sistema linear usado para treinamento.

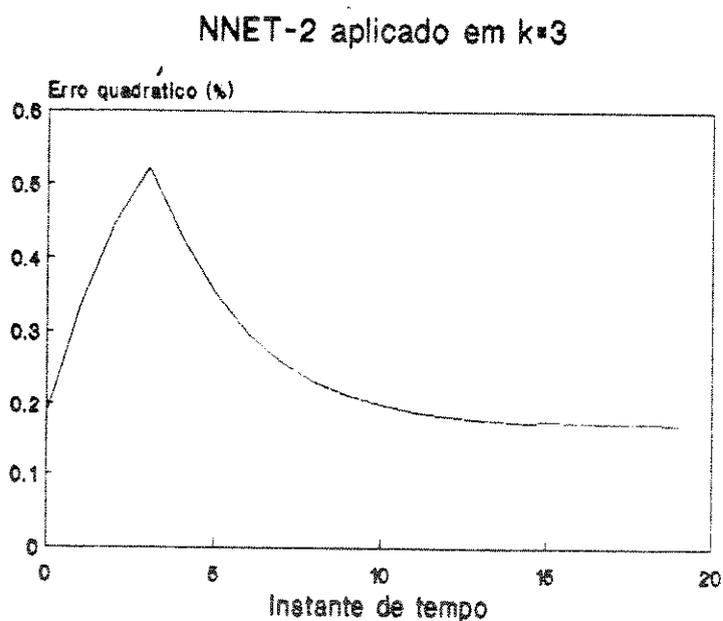


Fig. 10 - Desempenho da rede neural NNET-2 para linearização de um músculo estimulado eletricamente.

As simulações mostraram que a técnica apresentada é viável para o emprego no controle de músculos estimulados eletricamente. Nesta simulação foi suposto que os parâmetros do sistema não linear sejam fixos. O emprego do método em um esquema on-line através do treinamento adaptativo conforme apresentado na fig. 8, deverá fazer parte de pesquisas futuras. Tal esquema somente é possível, com a atual tecnologia de redes neurais, para sistemas com variações lentas de parâmetros. Os casos de músculos sujeitos a fadiga são exemplos de tais sistemas e portanto, passíveis de linearização on-line. ■

Sistemas tempo contínuo são analisados nas seções seguintes onde esquemas de linearização por realimentação são propostos, incluindo uma extensão do método para aplicação em sistemas amostrados.

2. Linearização por Realimentação Através do Uso de Redes Neurais para de Sistemas de Tempo Contínuo Não Identificados

Mostrou-se no capítulo II que, se certas condições técnicas forem satisfeitas, um sistema não linear de tempo contínuo do tipo ψ (linear no controle) pode ser linearizado através de realimentação e transformação de coordenadas. Foi também analisada a linearização de sistemas amostrados, verificando-se que o método pode ser aplicado somente à uma classe restrita de sistemas. Mostrou-se também, na seção anterior, que a função de linearização por realimentação e a mudança de coordenadas podem ser aproximadas por redes neurais para o caso de sistemas de tempo discreto. Nesta seção, demonstra-se que a função de linearização por realimentação pode também ser aproximada por

uma rede neural, para o caso de sistemas do tipo ψ_N , que se constituem em uma classe mais geral de sistemas não lineares de tempo contínuo. Neste caso, a amostragem somente será realizada com a finalidade de treinamento. Uma vez que a rede neural esteja ajustada, não haverá mais necessidade de amostragem, caso seja usada uma rede neural composta de elementos analógicos, na implementação do controle de linearização por realimentação. Analisa-se na próxima seção a linearização por realimentação para sistemas do tipo ψ_s ou ψ_N .

2.1 - Linearização por realimentação de sistemas de tempo contínuo do tipo ψ_N

O primeiro passo para linearização por realimentação de sistemas de tempo contínuo, consiste na identificação da dinâmica inversa do sistema. No capítulo III, um método é apresentado, baseando-se no emprego de α derivadas da saída, onde α é a ordem relativa do sistema. Em vez de se usar derivadas da saída, pois poderão apresentar problemas de medição em sistemas industriais¹, escolhe-se um intervalo de amostragem e determina-se a dinâmica inversa baseada em α valores prévios de saída. A determinação do intervalo de amostragem e os procedimentos de treinamento das redes devem seguir as diretivas dos capítulos I e III. Algumas definições são necessárias.

¹.Pode-se também reconstruir derivadas da saída usando-se um filtro de variáveis de estado que aproxima um atraso de tempo puro.

A função inversa amostrada θ^{-1} (veja capítulo III, definição 1) é um mapeamento de $Y^{n+1} \rightarrow U$ tal que, $u(k\delta) = \theta^{-1}(y((k+1)\delta), y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta))$. A função inversa amostrada de um sistema de tempo contínuo ψ_s determina o valor da entrada amostrada $u(k\delta)$ (considerada constante dentro do intervalo de amostragem δ), que aciona o sistema ψ_s dos valores prévios de saída $y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta)$ para o valor $y((k+1)\delta)$. O inteiro n é a ordem do sistema amostrado de entrada-saída ψ_s , que poderá ser estimada através dos esquemas discutidos no capítulo III. Considerando-se que θ^{-1} seja suave, pode-se aproximá-la por uma rede neural de várias camadas sob uma tolerância τ , quando estiver disponível uma quantidade suficiente de amostras para representar o espaço de definição $Y^{n+1} \times U$.

Definição 2 - A função de linearização por realimentação

$$\gamma: \{ X \times V \rightarrow U \mid u(t) = \gamma(x(t), v(t)) \} \quad (07)$$

é um mapeamento contínuo que relaciona o estado $x(t) \in X$ e a entrada $v(t) \in V$ ao sinal atuante $u(t) \in U$. X é um conjunto aberto de \mathbb{R}^n , U e V são conjuntos abertos de \mathbb{R}^m . O sinal $u(t)$ é escolhido de tal forma que quando aplicado como entrada ao sistema ψ_N , a relação de entrada-saída $\{v(t), y(t)\}$ seguirá a relação de entrada-saída $\{v(t), y_L(t)\}$ do sistema linear do tipo \mathcal{L} (eq. II.10).

Proposição 4: Uma rede neural com pelo menos uma camada intermediária, poderá ser treinada para aproximar a função γ que lineariza por realimentação o sistema ψ_N , considerando-se que os seguintes dados estejam disponíveis para treinamento.

Saída:

$$u(k\delta) = \theta^{-1}(y_L((k+1)\delta), y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta)) \quad (08)$$

isto é, a dinâmica inversa do sistema ψ_N , acionada pela saída $y_L((k+1)\delta)$ do sistema linear \mathcal{L} , usado como modelo, e as saídas prévias do sistema ψ_N :

$$\{y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta)\}.$$

Assim sendo, $u(k\delta)$ é o sinal atuante que aciona ψ_N dos valores prévios $\{y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta)\}$ para $y_L((k+1)\delta)$.

Entrada: A mesma entrada $v(k\delta)$ do sistema linear \mathcal{L} e o estado $x(k\delta)$ do sistema ψ_N no instante $k\delta$.

Demonstração: Desde que γ seja uma função suave, pode-se aproximá-la por uma rede neural com pelo menos uma camada intermediária como mostrado no capítulo I, considerando-se que um número suficiente de amostras de definição dos espaços $X \times V$ e U esteja disponível para treinamento da rede.

Embora se esteja tratando de sistemas de tempo contínuo, a amostragem é usada somente para fins de treinamento da rede neural para aproximar a função suave γ . Portanto, durante o treinamento, o sistema é discretizado e as mesmas considerações da proposição 3 se aplicam, neste caso. ■

Para se ter uma aproximação razoável, as diretivas apresentadas no capítulo I, para treinamento da rede neural, devem

ser aplicadas usando-se diferentes intervalos de tempo para garantir que a tolerância desejada seja atingida.

Embora esteja sendo usada uma realimentação de estado para o treinamento da rede neural, o sistema não se torna linear em relação à resposta entrada-estado. A função γ captada pela rede neural, somente fornece o controle u que aciona a saída do sistema a partir dos valores prévios de saída $\{y(k\delta), y((k-1)\delta), \dots, y((k-n+1)\delta)\}$ para $y_L((k+1)\delta)$. Portanto, o estado do sistema não-linear não segue necessariamente o estado do sistema linear. Somente a função de linearização por realimentação $\gamma(x,v)$ não será suficiente para linearizar o sistema da entrada para o estado. Uma mudança de coordenadas será também necessária para transformar o estado x no estado do sistema linear z .

Os procedimentos de linearização por realimentação no espaço de estado poderão ser resumidos nos seguintes passos:

Fase 1: Treinamento da dinâmica inversa da rede neural NNET-1 como ilustrado na fig. 11.

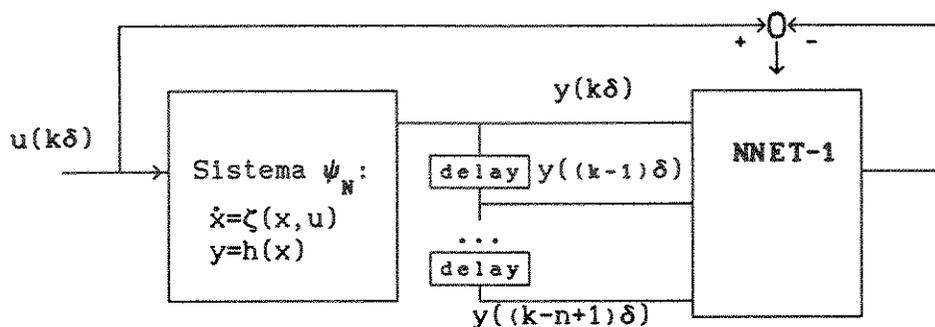


Fig. 11 - Treinamento de NNET-1 com a dinâmica inversa do sistema de tempo contínuo ψ_N usando-se dados amostrados.

Fase 2: Treinamento da rede neural de linearização por realimentação (fig. 12). Note que o sistema é amostrado somente para fins de treinamento da rede neural.

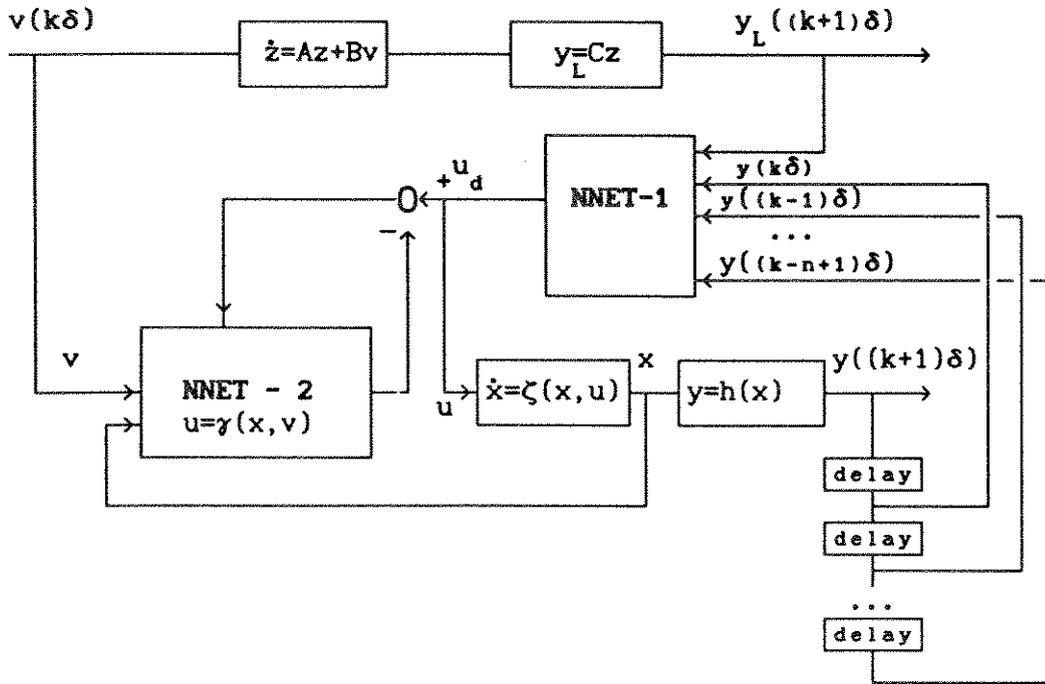


Fig. 12 - Treinamento da rede de linearização por realimentação NNET-2.

O treinamento neste caso, deve ser executado para tempos de amostragem diferentes, para assegurar que NNET-2 tenha uma razoável capacidade de generalização. Note que NNET-1 e a ordem no sistema amostrado n dependem do intervalo de amostragem escolhido. Portanto, para se mudar o intervalo de amostragem, na fase 2, tem-se que voltar a fase 1 e retreinar NNET-1, usando-se as diretivas do capítulo III.

Fase 3: Nesta fase a transformação de coordenadas da rede neural é treinada como ilustrado na fig. 13.

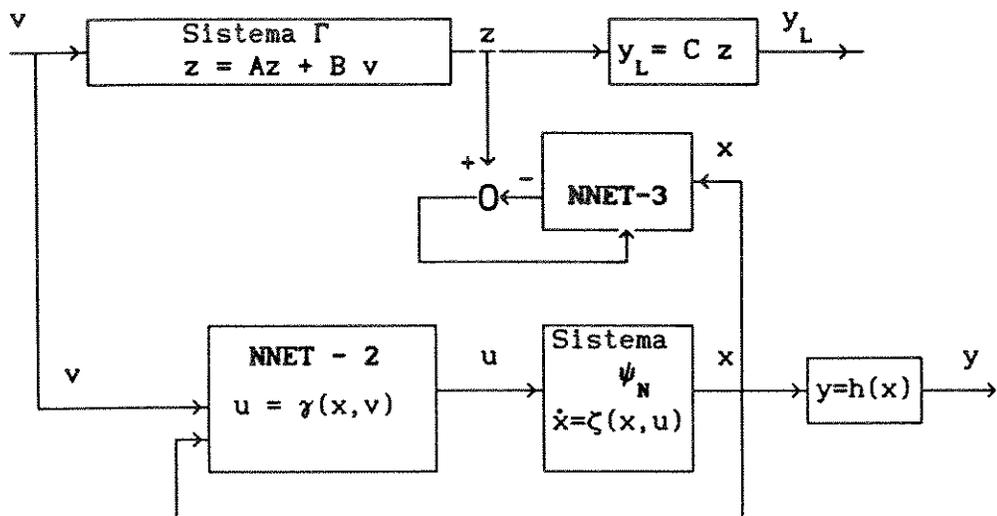


Fig.13 - Treinando a rede neural para transformação coordenadas.

Para se treinar NNET-3, é novamente necessário se amostrar o sistema. Note que NNET-2, já treinado, é usado como uma realimentação de tempo contínuo, e o sistema se torna linear de v para y .

Fase 4: Na operação não-adaptativa do sistema, NNET-2 e NNET-3, já treinadas, são usadas continuamente, e o sistema se comporta linearmente da entrada para a saída ($v-y$) e da entrada para o estado ($v-z$). A figura 14 ilustra esta operação.

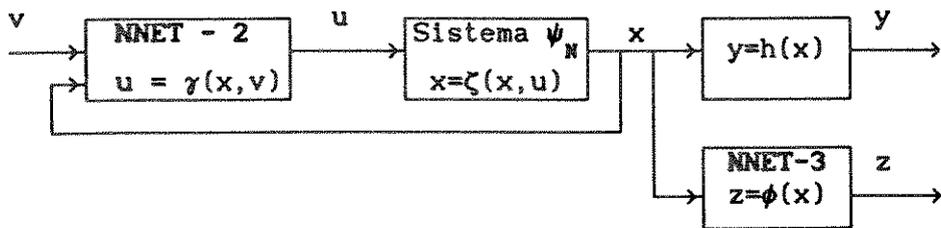


Fig.14- Operação normal: linearização por realimentação de entrada-saída-estado.

Treinamento adaptativo não é possível, neste caso, pois as redes analógicas NNET-2 e NNET-3 ficam continuamente conectadas ao sistema. Ajuste adaptativo dos pesos das redes somente é possível para sistemas de tempo discreto com variações lentas, onde algumas iterações de propagação retroativa podem ser aplicadas durante os intervalos de amostragem.

Note também que este método não consiste em linearização por realimentação através de amostragem, pois a amostragem somente é usada para treinar as redes neurais. Portanto os efeitos de amostragem em linearização por realimentação, discutidos no capítulo II, não se aplicam neste caso.

Considerando-se que o uso de redes neurais é um método de aproximação, o resultado final é uma aproximação do sistema linear desejado. Deve-se considerar também que esta aproximação é somente válida para o espaço de definição usado no treinamento da rede neural. Embora as redes neurais possam generalizar para pontos não usados na fase de treinamento, deve-se levar em conta que pontos fora do espaço de definição poderão resultar em generalizações imprecisas.

O método discutido nesta seção se aplica a sistemas do tipo ψ_M que se constituem em uma classe mais geral do que os sistemas do tipo ψ . Linearização por realimentação de sistemas do tipo ψ se encontra extensivamente discutida na literatura especializada (Isidori, 1989, Sastry, 1990 e capítulo II). A linearidade da entrada u permite a aplicação de um rigoroso tratamento matemático e um método analítico para a síntese da função de linearização e da transformação de coordenadas. Considerando-se que o sistema ψ é uma subclasse do sistema ψ_M , o método apresentado neste capítulo se aplica também aos sistemas tipo ψ . Assim sendo, o método aqui apresentado não exige que se leve em consideração a estrutura particular da função de linearização e não está restrito à classe particular de sistemas, denominados analíticos-lineares.

Embora este método seja aplicável a sistemas não lineares não identificados, algum conhecimento prévio é necessário, tal como a ordem do sistema e o grau relativo. A ordem do sistema linear usado como modelo deverá ser igual ao grau relativo do sistema não linear original (veja capítulo II).

2.2 - Linearização por realimentação de sistemas de tempo contínuo do tipo ψ_s (sem acesso ao estado)

Para o caso de sistemas amostrados, passíveis de linearização por realimentação porém sem acesso ao estado, pode-se alimentar a rede NNET-2 com n amostras passadas de saída do sistema. Desta forma, a rede será treinada, a partir dos dados amostrados e somente poderá ser aplicada se o mesmo intervalo de amostragem, usado para treinamento, for também aplicado durante a fase de operação normal da rede.

O exemplo seguinte ilustra a linearização por realimentação em sistemas de tempo contínuo, mostrando que resultados bastante precisos são obtidos, mesmo para o caso de sistemas amostrados.

Exemplo 2: Neste exemplo, apresenta-se a simulação da fase de treinamento e o teste de desempenho da rede de linearização por realimentação, para o mesmo motor cc, cuja dinâmica inversa foi identificada no Capítulo III (exemplo 1). Três casos são apresentados: nos casos 1 e 2 mostra-se o desempenho das redes na identificação das funções de linearização por realimentação de entrada-saída; no terceiro caso apresenta-se também os resultados da identificação da transformação de coordenadas para linearização no espaço de estados.

Caso 1: Neste caso usou-se a rede NNET-1 treinada anteriormente com a dinâmica inversa de um motor cc, apresentada no Cap. II (Exemplo 1, Caso 1). O esquema apresentado na figura 12 foi empregado no treinamento da rede NNET-2, com exceção que valores de saída foram alimentados à rede. Durante esta fase, apresenta-se a entrada de NNET-2 a tensão $v(t)$ e a realimentação da velocidade de saída do motor. Um sistema linear, tomado como modelo, é simulado simultaneamente recebendo como excitação o mesmo valor de tensão que é introduzido na rede NNET-2, a ser treinada. A saída dos sistemas linear e não linear são introduzidas na rede NNET-1. Conforme demonstrado na Proposição 4, a saída de NNET-1 corresponde ao valor de tensão de armadura capaz de fazer com que a velocidade do motor se comporte de forma linear seguindo o modelo apresentado. Assim sendo, NNET-2 é treinada a apresentar em sua saída o valor equivalente a saída de NNET-1. Usou-se como modelo linear o seguinte sistema, embora qualquer modelo estável pudesse ser empregado:

$$\frac{dz}{dt} = -z + v \quad (09)$$

onde z , $v \in \mathbb{R}$, são a variável de saída e de excitação, respectivamente e t é a variável independente.

Após 7500 apresentações, foi atingido um mínimo local do erro quadrático igual a 0.0014%, calculado a cada intervalo de 1000 apresentações (vide fig. 15). A rede NNET-2 foi modelada com 50 elementos em sua camada intermediária.

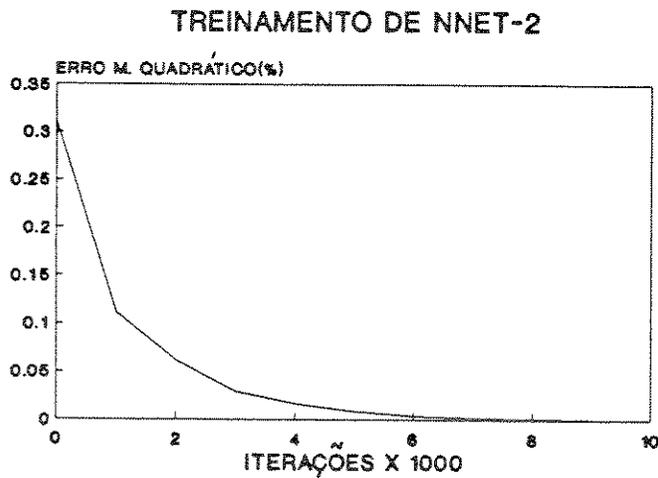


Fig. 15 - Treinamento da rede NNET-2, com a função de linearização de um motor cc, com $\lambda(\omega) = \text{sen}(0.1\omega)$.

Após o treinamento, a rede NNET-1, responsável pela dinâmica inversa, foi desconectada e o sistema foi simulado tendo a rede NNET-2 como controle. A rede NNET-2 recebe a tensão de controle $v(t)$ em sua entrada, além de uma realimentação correspondente a velocidade do motor. A sua saída $u(t)$ foi conectada à entrada de tensão da armadura do motor. O sistema linear (09) foi também simulado com a excitação $v(t)$ e sua saída $z(t)$ foi comparada com a velocidade angular $\omega(t)$ do motor. O erro médio quadrático desta comparação está apresentado na fig. 16.

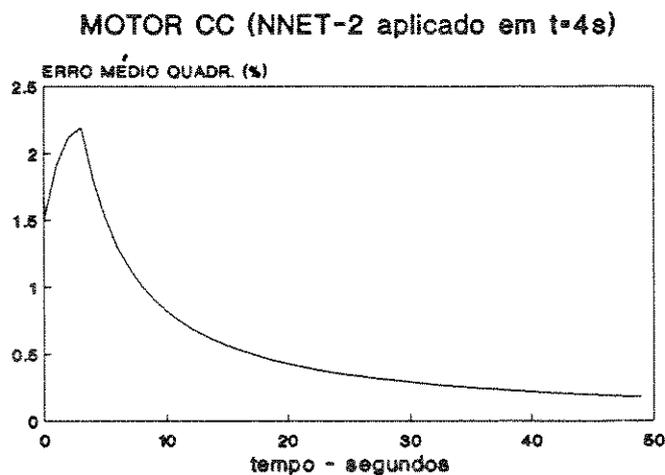


Fig. 16 - Teste de desempenho da rede NNET-2 responsável pela linearização por realimentação.

Para mostrar o efeito de linearização da rede NNET-2, o motor foi alimentado até $t=4s$ com $v(t)$, quando se passou a excitá-lo com a saída $u(t)$ de NNET-2. Como se pode observar pelo crescimento do erro médio geométrico de 0 a 4 segundos, os dois sistemas estavam se distanciando. Após a alimentação do motor com $u(t)$ os dois sistemas passaram a apresentar praticamente a mesma saída, como se pode observar pelo decréscimo do erro médio quadrático (fig. 17) e pelo gráfico da figura 18 onde são plotados $z(t)$ e $\omega(t)$. A partir de 4 seg. as curvas ficaram praticamente coincidentes.

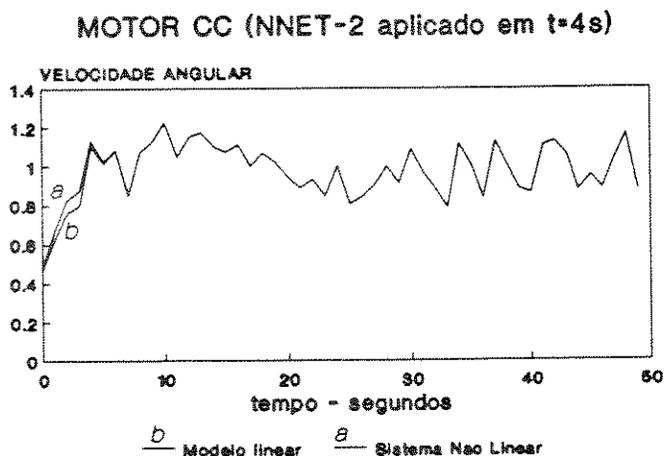


Fig. 17 - Velocidade do motor cc versus saída do sistema linear

Na figura 17 estão plotados a entrada $v(t)$, que é aplicada tanto ao sistema linear como a entrada da rede NNET-3, e o sinal atuante de $u(t)$.

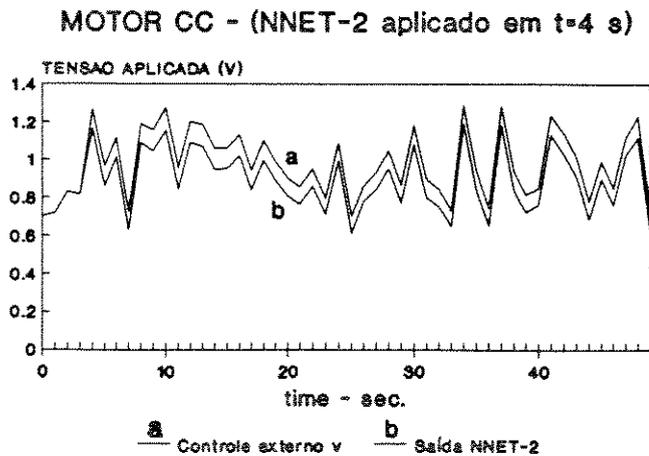


Fig. 18 - Excitação $v(t)$ versus saída de NNET-2 $u(t)$.

Os resultados obtidos neste exemplo ilustram a eficiência do método proposto para a identificação da função de linearização. Embora no caso deste exemplo as equações do sistema fossem conhecidas, estas não foram usadas para identificação da função de linearização, mas somente para gerar amostras de entrada e saída do sistema. Em um sistema industrial estas amostras devem ser medidas diretamente do sistema físico e usadas on-line no treinamento da rede neural. Salienta-se também que, as limitações levantadas por Grizzle (1988) e Jakubczyk, com relação a

linearização por realimentação de sistemas amostrados (vide cap. II), não se aplicam neste caso. Isto se deve ao fato de que a amostragem somente é usada para treinamento da rede, não para implementação um esquema de linearização de um sistema de tempo contínuo, onde a função de linearização tenha sido determinada analiticamente, como é o caso dos sistemas analisados por Grizzle (1989) e Jakubczyck (1987). No método apresentado neste trabalho, a rede neural já treinada, aproxima de forma bastante precisa a função real, e a amostragem não se torna mais necessária, para os casos onde seja possível a implementação da rede de forma analógica. Por outro lado, para os casos em que a implementação da rede deve também ser feita de forma amostrada, como no caso do exemplo anterior, também se obtém uma estimativa bastante precisa da função de linearização. Observe que a rede assimila, não a função de linearização teórica, mas a função que faz com que a saída do sistema não linear se comporte como a saída do sistema linear usado como modelo para o período de amostragem especificado.

Para ilustrar como a não linearidade do sistema original influi no treinamento da rede, o mesmo sistema foi simulado com o torque da carga alterado. Os resultados desta simulação são apresentados a seguir:

Caso 2: Neste caso, o mesmo sistema do caso 1 é simulado empregando-se a função $\lambda(\omega) = \text{sen}(0.8\omega)$ como torque da carga. Usou-se a rede NNET-1 treinada anteriormente com a dinâmica inversa do motor cc, e que foi apresentada no Cap. III (Exemplo 1, Caso 2). O esquema apresentado na figura 12 foi também empregado na fase de treinamento da rede NNET-2. O mesmo sistema linear do caso 1 foi tomado como modelo.

Após 8000 apresentações foi atingido um mínimo local do erro quadrático igual a 0.0011%, calculado a cada intervalo de 1000 apresentações (vide fig. 19). A rede NNET-2 foi modelada, neste caso com 100 elementos em sua camada intermediária.

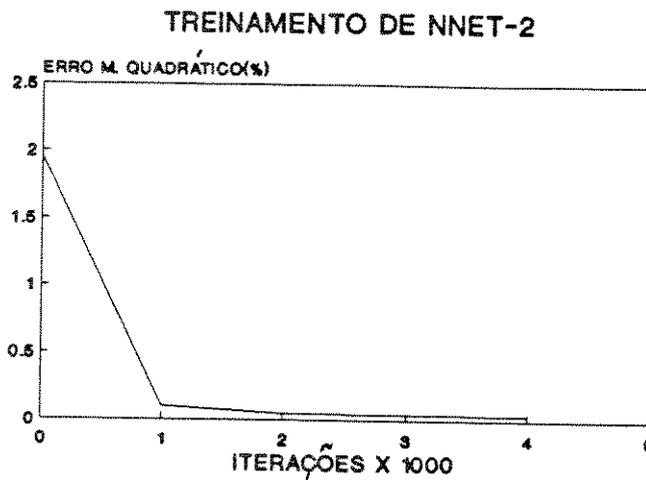


Fig. 19 - Treinamento da rede NNET-2, com a função de linearização de um motor cc, com $\lambda(\omega) = \text{sen}(0.8\omega)$.

Após o treinamento, a rede NNET-1, responsável pela dinâmica inversa, foi desconectada e o sistema foi simulado tendo a rede NNET-2 como controle. A tensão de controle $v(t)$ foi

introduzida na rede NNET-2, juntamente com uma realimentação correspondente a velocidade do motor. A saída $u(t)$ da rede NNET-2 foi conectada à entrada de tensão da armadura do motor. O sistema linear (09) foi também simulado com a excitação $v(t)$ e sua saída $z(t)$ foi comparada com a velocidade angular $\omega(t)$ do motor. O erro médio quadrático desta comparação está apresentado na fig. 20.

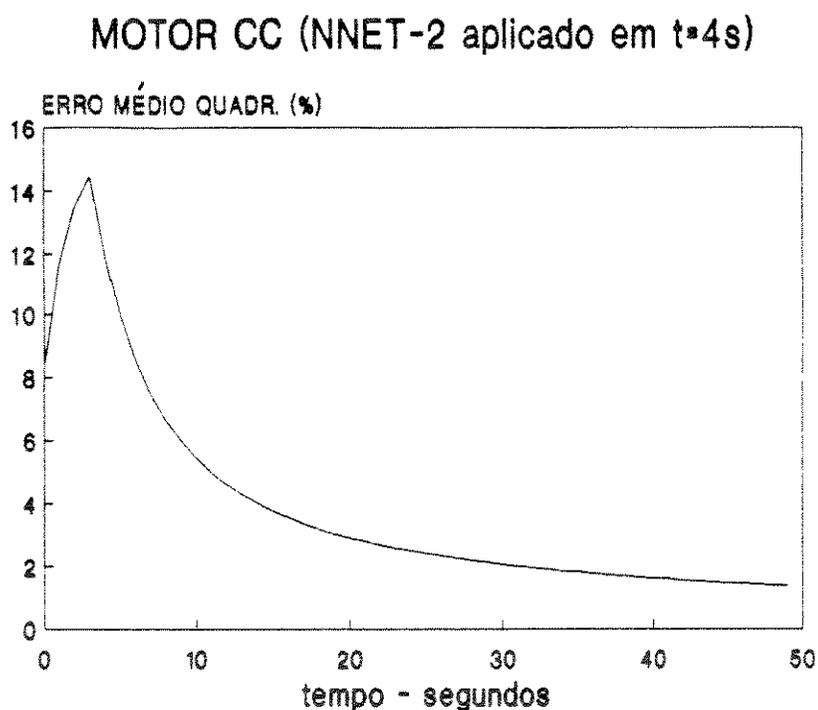


Fig. 20 - Teste de desempenho da rede NNET-2 responsável pela linearização por realimentação - NNET-2.

Assim como no caso anterior, o motor foi alimentado até

$t=4s$ com $v(t)$, quando se passou a alimentá-lo com a saída $u(t)$ de NNET-2. Como se pode observar pelo crescimento do erro médio geométrico de 0 a 4 segundos, os dois sistemas estavam se distanciando de forma mais acentuada que no Caso 1. Após a alimentação do motor com $u(t)$ os dois sistemas passaram a apresentar praticamente a mesma saída, porém de forma não tão precisa como no Caso 1. Em casos práticos, se a precisão atingida não satisfizer os padrões exigidos deve-se aumentar o número de elementos na camada intermediária para se melhorar a capacidade de generalização da rede. Como se pode observar pelo decréscimo do erro médio quadrático (fig. 21) e pelo gráfico da figura 22 (onde são plotados z e ω), os dois sistemas se comportaram de forma bastante semelhante.

MOTOR CC (NNET-2 aplicado em $t=4s$)

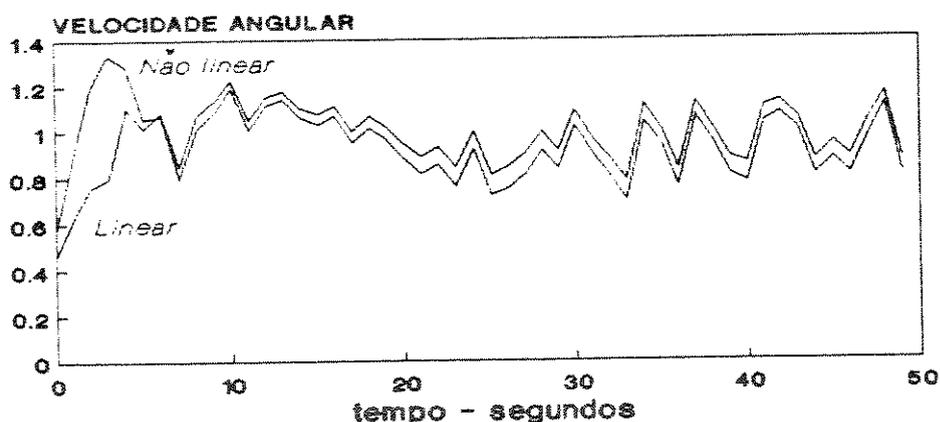


Fig. 21 - Velocidade do motor cc versus saída do sistema linear

Para ilustrar o efeito de NNET-2, a figura 22 mostra a comparação entre a entrada $v(t)$ (aplicada tanto ao sistema linear como a entrada da rede NNET-2) e o sinal atuante $u(t)$.

MOTOR CC - (NNET-2 aplicado em $t=4$ s)

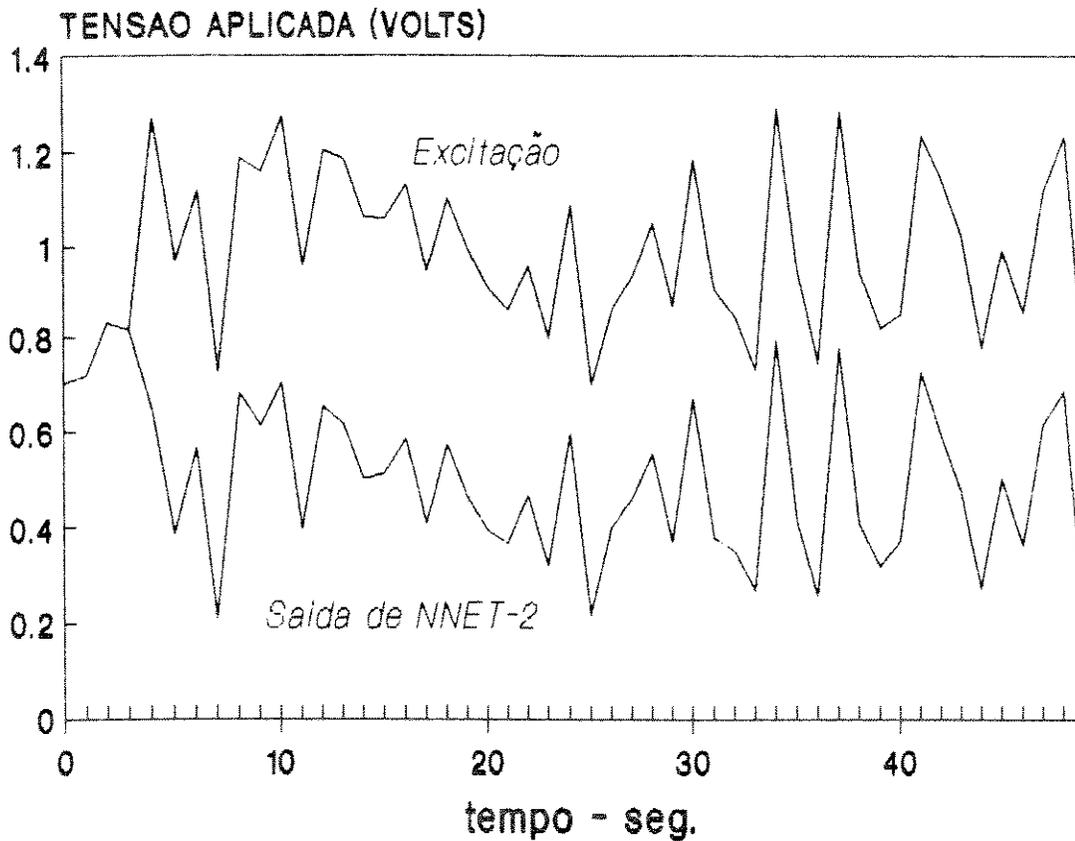


Fig. 22 - Excitação $v(t)$ versus saída de NNET-2 $u(t)$.

Analisando-se os casos 1 e 2 se conclui que o método apresentado foi eficaz para a estimação da função de linearização de entrada-saída por realimentação. No caso seguinte, o modelo de estado do motor cc é usado para linearizar o sistema no espaço de estado, através da estimação da transformação de coordenadas pela rede neural NNET-3.

Caso 3: Neste caso, foi simulado o mesmo exemplo apresentado no Cap. III (Ex. 1, caso 3) onde o modelo completo do motor CC foi simulado e sua dinâmica inversa foi identificada pela rede NNET-1.

Conforme demonstrado na sessão 2.1 deste capítulo, três fases são necessárias para a linearização no espaço de estado. Na primeira fase estima-se a dinâmica inversa do sistema pela rede NNET-1 (cap. III). Na segunda fase usa-se a rede NNET-1 para estimar a função de linearização de entrada-saída, treinando-se a rede NNET-2. Na terceira fase, a rede NNET-1 não é mais usada, e a rede NNET-3 é treinada com a transformação de coordenadas capaz de linearizar o sistema no que se refere a resposta entrada-estado.

Usando-se a rede NNET-1, já treinada e apresentada no Exemplo 2 (caso 3, cap. III), treinou-se a rede NNET-2 na identificação da função de linearização de entrada-saída. O seguinte sistema linear foi usado como modelo:

$$\frac{d\omega}{dt} = i + \omega \quad (10a)$$

$$\mu \frac{d\omega}{dt} = -\omega - i + u, \quad \mu = T_e / T_m \quad (10b)$$

onde: ω - velocidade angular do motor,
 i - corrente na armadura,
 u - tensão aplicada na armadura,
 T_e - Constante de tempo elétrica,
 T_m - Constante de tempo mecânica.

Na fig. 23 apresenta-se o desenvolvimento do erro médio geométrico durante a fase de treinamento da rede. Como se pode

observar, foram necessárias 8000 sessões de treinamento para se atingir um mínimo local de 0.0012. Também neste caso foi empregada uma rede neural com uma única camada intermediária com 100 elementos.

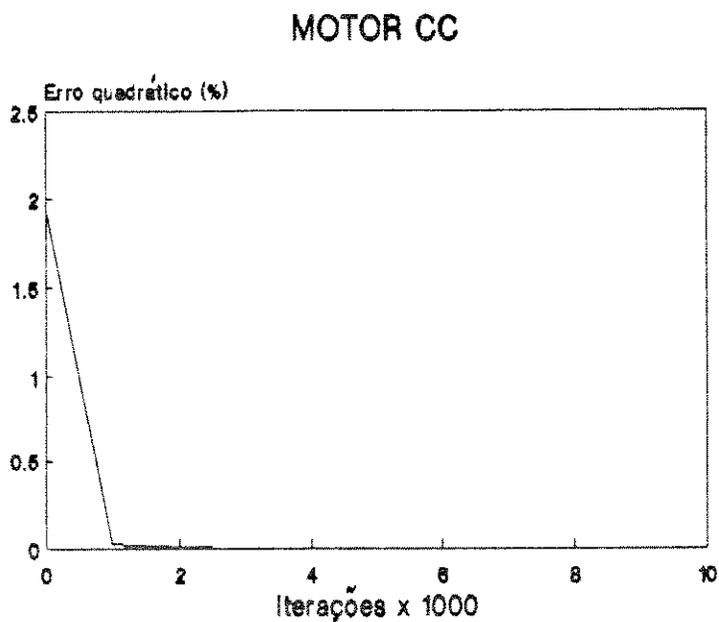


Fig. 23 - Treinamento de NNET-2 com a função de linearização de um motor CC.

Após o treinamento, a rede NNET-2 foi testada recebendo como entrada a tensão de controle $v(t)$ juntamente com uma realimentação correspondente a velocidade do motor. A saída $u(t)$ da rede NNET-2 foi conectada à entrada de tensão da armadura do

motor. O sistema linear (10) foi também simulado com a excitação $v(t)$ e sua saída $z(t)$ foi comparada com a velocidade angular $\omega(t)$ do motor. O erro médio geométrico desta comparação está apresentado na fig. 24.

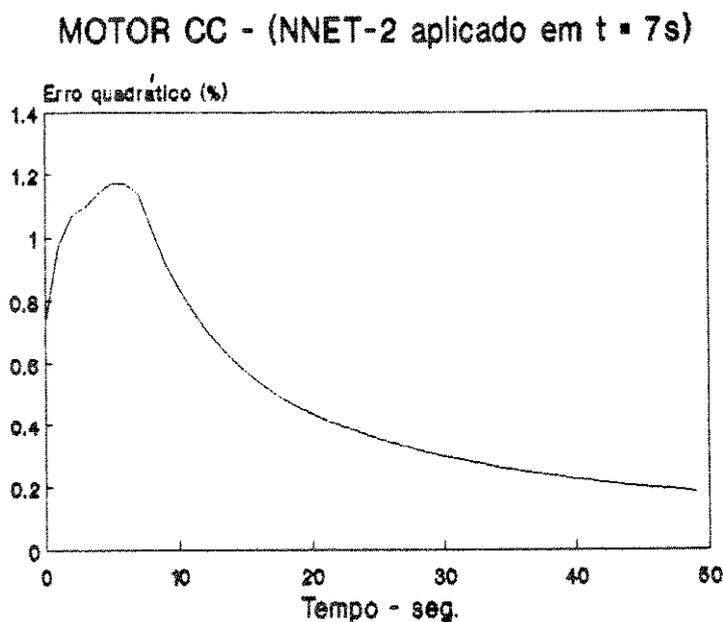


Fig. 24 - Teste de desempenho da rede NNET-2 responsável pela linearização por realimentação - NNET-2.

Assim como no caso anterior, o motor foi alimentado até $t=4s$ com $v(t)$, quando se passou a alimentá-lo com a saída $u(t)$ de NNET-2. Como se pode observar pelo crescimento do erro médio

geométrico, de 0 a 4 segundos os dois sistemas estavam se distanciando e, após a conexão de NNET-2, passaram a se comportar de forma bastante semelhante. Esta afirmação é verificada pelo decréscimo do erro médio geométrico (fig. 24) e pelo gráfico da figura 25 onde são plotados $z(t)$ e $\omega(t)$.

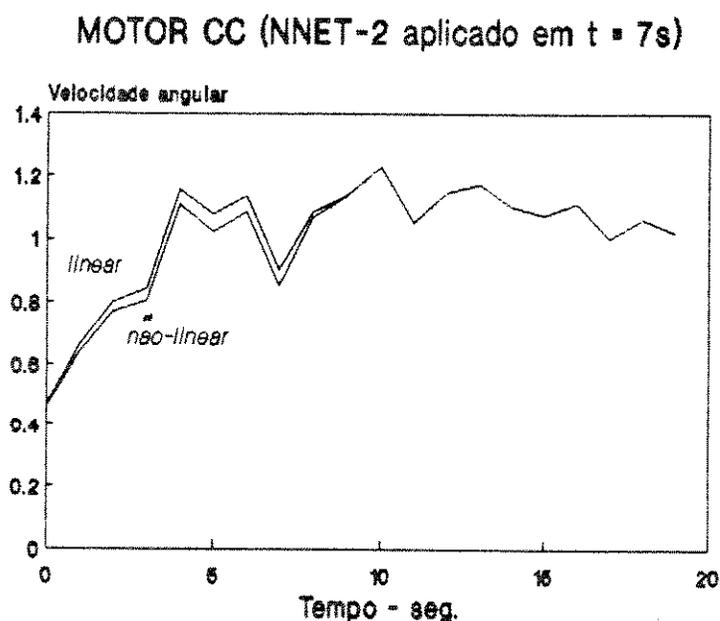


Fig. 25 - Velocidade do motor cc versus saída do sistema linear

Na figura 26 estão plotados a entrada $v(t)$ (aplicada tanto ao sistema linear como a entrada da rede NNET-2) e o sinal atuante de $u(t)$.

MOTOR CC - (NNET-2 aplicado em t=7s)

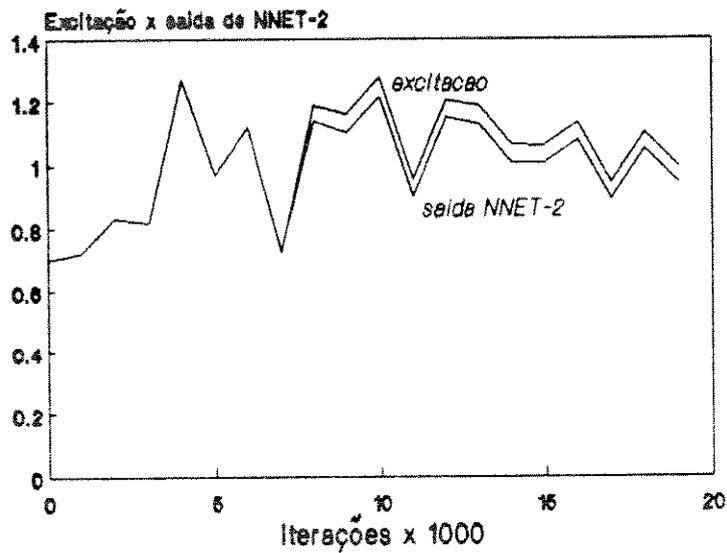


Fig. 26 - Excitação $v(t)$ versus saída $u(t)$ da rede NNET-2.

Após o treinamento de NNET-2, o esquema da fig. 13 foi usado no treinamento da rede NNET-3, para identificar a transformação de coordenadas que lineariza o sistema em termos da resposta entrada-estado. A rede NNET-2 já treinada foi usada para linearizar o sistema com relação a resposta entrada-saída. Durante o treinamento, o estado do sistema não linear foi introduzido na rede NNET-3, que foi treinada a apresentar em sua saída o estado

do sistema linear usado como modelo. Foram necessárias 11000 sessões de treinamento para se atingir um erro médio geométrico de 0.17%, como se pode observar na fig. 27.

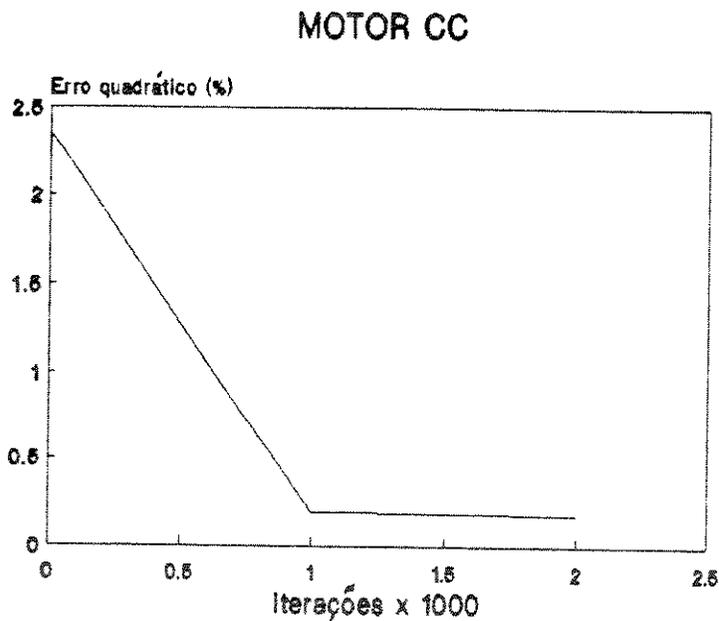


Fig 27 - Treinamento de NNET-3 com a transformação de coordenadas para linearização no espaço de estados.

Após o treinamento, o sistema foi simulado novamente com as rede NNET-2 e NNET-3 em operação e as saídas de NNET-3 foram

comparadas com o estado do sistema linear usado como modelo. O erro médio geométrico desta comparação está apresentado na fig. 28.

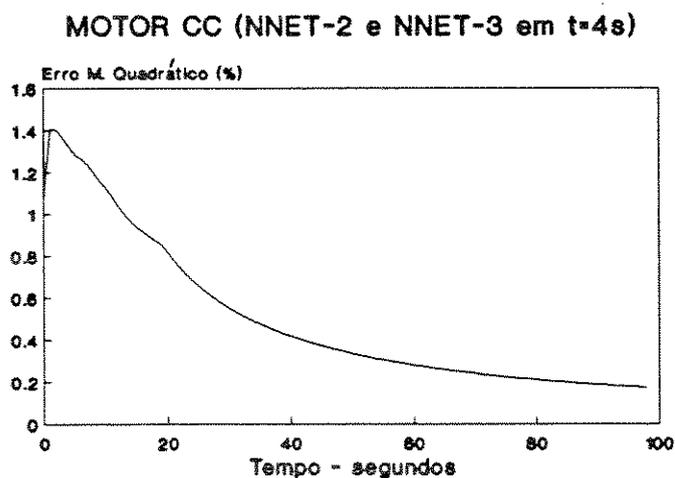


Fig. 28 - Desempenho da rede NNET-3, na transformação de coordenadas para linearização entrada-estado.

Os casos simulados mostram que através do método apresentado neste trabalho, pode-se obter tanto a função de linearização de entrada saída como a transformação de coordenadas

para a linearização no espaço de estado. O método se aplica principalmente para sistemas não identificados ou para sistemas onde a obtenção da função de linearização é matematicamente impraticável. Como ficou claro pela comparação dos casos 1 e 2, sistemas com alto grau de não linearidade são mais difíceis de serem treinados e exigem redes neurais de maiores dimensões. Para tanto, é aconselhável um certo conhecimento do sistema físico em questão, embora não seja necessário um perfeito equacionamento matemático do sistema, visto que as redes neurais são treinadas com base nas amostras de entrada-saída.

3. Conclusão

Foi proposto neste capítulo, um método para se controlar uma determinada classe de sistemas não lineares usando-se basicamente 3 redes neurais na fase de treinamento. A primeira rede neural que realiza a dinâmica inversa do sistema somente é usada para treinar a segunda rede neural que assimila a regra de controle desejada. Considerando-se que o sistema inverso não é usado no controlador propriamente dito, mas somente na fase de treinamento, espera-se que o sistema assim controlado apresente um desempenho robusto em malha fechada. Os resultados das simulações mostraram que o método é aplicável para sistemas de tempo contínuo e de tempo discreto. Para sistemas de tempo discreto, um método adaptativo é apresentado, permitindo que sistemas com variações lentas de parâmetros sejam adaptativamente linearizados. O método também se mostrou eficaz nos testes com sistemas contínuos amostrados.

Conclusão Geral

Apresentou-se inicialmente, neste trabalho, um método para se avaliar do número de amostras e o número de unidades das camadas intermediárias em uma arquitetura de rede neural, para aproximação funcional. Foram também apresentadas considerações com relação ao treinamento on-line de redes neurais conectadas a sistemas dinâmicos.

O método analítico para linearização por realimentação de sistemas não lineares foi analisado, com vistas a se estabelecer uma estrutura analítica para servir de base de comparação com o método desenvolvido neste trabalho, isto é, linearização por realimentação usando-se redes neurais.

Considerando-se que o método proposto exige a inversão de sistemas dinâmicos, este assunto foi analisado em termos dos métodos convencionais. Em seguida, foi apresentado um método para se determinar a dinâmica inversa de sistemas não lineares através de redes neurais. O método pode tanto ser aplicado para sistemas de tempo discreto como contínuo. Para o caso de tempo contínuo, foi apresentado um método alternativo para se evitar medições de derivadas da saída. O desempenho das redes foi testado através de exemplos e os resultados das simulações mostraram que a generalização desejada foi atingida.

Como principal objetivo, foi apresentado um método para se controlar sistemas não lineares usando-se redes neurais. Embora o método faça uso da dinâmica inversa do sistema, esta somente é usada para treinar a segunda rede neural que assimila a regra de

controle desejada. Considerando-se que o sistema inverso não é usado no controlador propriamente dito, mas somente na fase de treinamento, espera-se que o sistema assim controlado apresente um desempenho robusto em malha fechada. O método se mostrou eficaz, tanto para sistemas de tempo contínuo como discreto. Para sistemas de tempo discreto, um método adaptativo é apresentado, permitindo que sistemas com variações lentas de parâmetros sejam adaptativamente linearizados. O método também se mostrou eficaz nos testes com sistemas contínuos amostrados.

Considerando-se que o método apresentado neste trabalho, não faz uso da estrutura convencional do método analítico de linearização por realimentação, espera-se que as restrições do método convencional não se apliquem ao método aqui apresentado. Esta questão carece de uma investigação matemática e deverá fazer parte de trabalhos futuros.

APÊNDICE I

Redes Neurais Artificiais de Várias Camadas

1. Introdução

Desde o desenvolvimento do Perceptron na década de 1950, várias tentativas foram feitas de se imitar o desempenho do cérebro humano através de dispositivos capazes de aprender por treinamento. Entretanto, somente a partir da década de 1980, resultados realmente positivos foram obtidos.

Estes dispositivos foram denominados redes neurais artificiais e podem ser classificados, segundo o método de treinamento, em redes com aprendizagem supervisionada e não supervisionada. Considerando-se que neste trabalho se concentrou nas aplicações de redes com aprendizagem supervisionada, discute-se neste apêndice os detalhes matemáticos do método de treinamento de tais redes. Em Rumelhart(1987) se encontra uma completa discussão das redes de várias camadas e em Lippmann(1987), diversos tipos de redes, supervisionadas ou não supervisionadas, são descritas. Neste apêndice, procura-se detalhar a formulação matemática das redes de várias camadas, visto que constituem a principal ferramenta utilizada neste trabalho, para a aproximação funcional de mapeamentos não lineares.

Dentre as redes não supervisionadas se destacam a rede de Hopfield e a rede Kohonen. A rede de Hopfield, é usada como memória associativa em tarefas de reconhecimento de padrões. Por se constituir em um sistema não linear dinâmico, o seu emprego como aproximação funcional fica bastante limitado. Esta rede tem sido bastante investigada, principalmente no que se refere a sua

capacidade como memória associativa (Lippmann,1987, Hopfield,1987, Teixeira,1989), e na solução de problemas de otimização.

Basicamente, as redes neurais são constituídas de vários elementos interconectados cujas funções básicas podem ser descritas pela equação (1).

$$y = f\left(\sum_{j=1}^n w_j x_j - \theta\right) \quad (01)$$

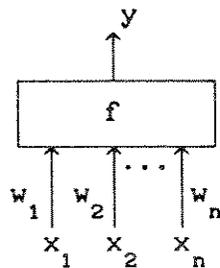
onde: y - saída do elemento;

$x \in \mathbb{R}^n$ - entrada do elemento;

$w \in \mathbb{R}^n$ - pesos de conexão;

f - função de ativação;

θ - valor de limiar do elemento.



ig. 1 - Um elemento de uma rede neural artificial

Diversos tipos de funções podem ser usadas como função de ativação, tais como: limitadores bruscos, ativação linear ou funções sigmóides. A função sigmóide é geralmente usada em redes de várias camadas por se tratar de uma função monótona e crescente.

As primeiras redes neurais eram constituídas de uma camada de entrada e uma camada de saída. Tais redes são impróprias para aproximação funcional e são bastante limitadas para uso em

reconhecimento de padrões. Embora tenham pouco interesse prático, as redes de uma camada são descritas a seguir, por servirem de base para as redes de várias camadas.

2. Redes Neurais de Uma Camada

Em uma rede de uma camada, o conjunto de entrada $x \in \mathbb{R}^{N_e}$ é relacionado ao conjunto de saída $y \in \mathbb{R}^{N_s}$, através de vários elementos descritos pela equação (1). A matriz de pesos das conexões W de dimensões $N_s \times N_e$ deve ser ajustada de forma a satisfazer um conjunto de m padrões de entrada-saída. As aplicações destas redes são bastante limitadas, pois são somente capazes de identificar padrões ortogonais entre si.

Basicamente, o treinamento de uma rede neural é feito através da modificação do conteúdo da matriz de pesos W . Este treinamento pode ser efetuado por diversos algoritmos e representa uma área de intensa investigação por parte de diversos pesquisadores. Especificamente, neste trabalho será analisada a regra delta generalizada, ou algoritmo de propagação retroativa (Rumelhart, 1987), para o treinamento de rede de várias camadas. Entretanto, para haver uma perfeita compreensão desse algoritmo, torna-se necessário uma análise da regra delta linear que foi inicialmente usada para treinamento das redes de uma camada.

3. Regra Delta Linear

Suponha uma rede neural onde todos os seus elementos possuam funções de ativação lineares. Pode-se verificar (Rumelhart, 1987) que esta rede somente terá a capacidade de identificar padrões ortogonais entre si. Suponha, então que os vetores apresentados à entrada da rede, na fase de treinamento, tenham norma igual a 1, além de serem ortogonais entre si. Segundo

a regra de Hebb (Rumelhart, 1987), a matriz de pesos será dada pelo produto exterior dos vetores de entrada e de saída. A matriz final será obtida pela soma das diversas matrizes obtidas com a apresentação dos diversos pares de entrada-saída. Considere:

- x_i - valor apresentado ao nó de entrada i ;
- s_j - valor desejado no nó de saída j ;
- w_{ij} - peso da conexão entre a entrada i e o elemento j .
- Δw_{ij} - variação do peso da conexão $i \rightarrow j$, a cada apresentação.

Considere também ΔW , como a variação causada pela adição de cada novo padrão (p) de entrada-saída, isto é:

$$\Delta W^{(p)} = s x^T, \quad s \in \mathbb{R}^{N_s}, \quad x \in \mathbb{R}^{N_e} \quad (02)$$

A matriz de pesos total será, portanto:

$$W = \sum_{p=1}^m \Delta W^{(p)} \quad (03)$$

onde m = número total de padrões ensinados a rede

Substituindo-se (2) em (3):

$$W = \sum_{p=1}^m s^{(p)} (x^{(p)})^T \quad (04)$$

Uma vez terminada a fase de treinamento, a rede poderá ser usada para reconhecer padrões apresentados a sua entrada, através da seguinte expressão:

$$y = f(W x) \quad (05)$$

onde W corresponde à matriz de pesos das conexões, com dimensões $N_s \times N_e$.

A regra delta linear é, na realidade uma derivação da regra de Hebb, exceto que utiliza o processo dos mínimos quadrados para ajuste da matriz de pesos, e pode ser sintetizada pela expressão (06):

$$W^{(p)} = W^{(p-1)} + \eta \delta^{(p)} (x^{(p)})^T \quad (06)$$

onde: $W^{(p)}$ - estado da matriz de pesos após a apresentação de p pares de entrada-saída;

$x^{(p)}$ - entrada da apresentação de ordem p;

η - escalar que determina a taxa de aprendizagem;

$\delta^{(p)}$ - diferença entre a saída atual e a desejada, tal que:

$$\delta^{(p)} = s - W^{(p-1)} x^{(p)} \quad (07)$$

$s^{(p)}$ - saída desejada na apresentação p;

$y^{(p)}$ - saída atual na apresentação p, sendo:

$$y^{(p)} = W^{(p-1)} x^{(p)} \quad (08)$$

O processo é iniciado zerando-se a matriz de pesos. A cada apresentação, a matriz é recalculada, pela expressão (06). Define-se desta forma as regiões no hiperespaço que correspondem às classes a serem definidas. Cada elemento da rede é capaz de dividir o hiperespaço em duas regiões separadas por um hiperplano.

Esta regra consiste basicamente na aplicação da técnica dos mínimos quadrados, de forma que, a cada apresentação, caminha-se em direção ao menor erro entre o valor atual e o valor de saída da rede. Pode-se comprovar esta afirmação mostrando-se que a derivada do erro em relação a cada elemento da matriz de pesos é proporcional a variação dos pesos calculada pela regra

delta, com uma constante de proporcionalidade negativa. Isto corresponde a se deslocar, na direção do gradiente no sentido de se atingir o erro mínimo. Seja:

$$E^{(p)} = \frac{1}{2} \sum_{j=1}^{Ns} (s_j - y_j)^2 \quad (09)$$

onde: $E^{(p)}$ - medida do erro quadrático de entrada-saída na apresentação p.

s_j - saída desejada na apresentação p, para o nó j.

y_j - saída calculada na apresentação p, para o nó j.

O erro total E é então dado por:

$$E = \sum_{p=1}^m E^{(p)} \quad (10)$$

Assim sendo, a variação na matriz de pesos, a cada apresentação, é dada por:

$$\Delta W_{j1} = \eta \delta_j x_1 \quad (11)$$

onde,

$$\delta_j = s_j - y_j \quad (12)$$

O treinamento da rede consiste portanto em se fazer várias apresentações dos pares de entrada-saída. Espera-se que após um certo número de apresentações, a rede tenha sido treinada, de forma que o erro se torne desprezível. O seguinte lema comprova que, a cada apresentação se caminha em direção ao erro mínimo.

Lema 1: A regra delta linear faz com que, a cada apresentação se caminhe em direção ao erro mínimo, pela direção do gradiente da função do erro quadrático, no espaço dos pesos das conexões.

Demonstração: Deve-se mostrar que ΔW é proporcional ao negativo da derivada do erro quadrático em relação aos pesos da conexões. Desta forma, demonstra-se que a cada passo o erro estará caminhando para o mínimo. Sendo assim, deve-se provar que:

$$-\frac{\partial E^{(p)}}{\partial W_{j1}} \propto \delta_j x_1 \quad (13)$$

Adotando-se uma constante de proporcionalidade η tem-se:

$$-\frac{\partial E^{(p)}}{\partial W_{j1}} = \eta \delta_j x_1 \quad (14)$$

A derivada poderá ser determinada por:

$$\frac{\partial E^{(p)}}{\partial W_{j1}} = \frac{\partial E^{(p)}}{\partial y_j} \frac{\partial y_j^{(p)}}{\partial W_{j1}} \quad (15)$$

da expressão (09) obtém-se:

$$\frac{\partial E^{(p)}}{\partial y_j} = (s_j - y_j) \quad (16)$$

ou,

$$\frac{\partial E^{(p)}}{\partial y_j} = -\delta_j \quad (17)$$

Como todos os elementos da rede são lineares pode-se deduzir pela equação (08) que:

$$\frac{\partial y_j}{\partial W_{j1}} = x_1 \quad (18)$$

Substituindo-se (17) e (18) em (15), obtém-se $-(\partial E^{(p)} / \partial W_{j1}) = \delta_j x_1$ que comprova a equação (14). ■

4. Rede de Várias Camadas e a Regra Delta Generalizada

Uma rede de várias de camadas (Rumelhart, 1987) apresenta a vantagem de não exigir que os padrões a serem ensinados sejam ortogonais entre si. Esta rede possui a capacidade de apreender estruturas bastante complexas e tem sido objeto de pesquisa por diversas áreas da ciência. Dentre suas aplicações se destacam: reconhecimento de imagens, sintetização e reconhecimento de voz, aplicações em previsão de mercado, problemas de otimização, e aplicações em sistemas de controle. Especificamente, no caso deste trabalho as redes neurais de várias camadas foram usadas para aproximar mapeamentos não lineares. Dentre as suas principais características pode-se citar a sua excelente capacidade de generalização, tanto no que se refere ao reconhecimento de padrões como em aproximação funcional. Foi demonstrado (Hornik, 1989) que estas redes são aproximadores universais e que somente uma camada intermediária é suficiente para aproximar qualquer mapeamento não linear.

Assim sendo a estrutura mínima da rede consiste em uma camada de entrada, com dimensão igual a dimensão do conjunto de entrada, uma camada intermediária, cujo número de elementos irá definir a treinabilidade e a precisão da aproximação funcional, e uma camada de saída com a dimensão do conjunto de saída.

Assim como na rede de uma camada, nas redes de várias camadas há a fase de treinamento e a fase de operação normal. Na fase de treinamento vários padrões de entrada-saída são apresentados à rede e os pesos das conexões são ajustados no sentido de minimizar o erro de saída. Na fase de operação normal os pesos não são mais alterados, sendo que os conjuntos são apresentados a entrada da rede e esta fornece os valores correspondentes em sua saída.

Pode-se verificar que a existência de camadas intermediárias só tem finalidade quando suas funções de ativação sejam não lineares. Uma rede linear de várias camadas só é capaz de assimilar o que uma rede linear de uma camada também é capaz de assimilar. Segundo Rumelhart(1987, vol2, pp65), foram estas limitações que levaram Minsky & Papert(1969) à avaliação pessimista da rede Perceptron. Infelizmente esta avaliação atingiu, na época, outras popostas de rede com grandes potencialidades.

A regra delta generalizada, desenvolvida por Rumelhart, trabalha com unidades não lineares e diferenciáveis e elimina praticamente todas as desvantagens das redes lineares. A função de ativação deve ser uma função diferenciável do tipo um para um, monótona e crescente. A função sigmóide, representada pela equação (19), satisfaz estas condições.

$$f(\alpha) = \frac{1}{1 + e^{-(\alpha+\theta)}} \quad (19)$$

onde α - representa a entrada líquida do elemento, isto é, o produto dos pesos pelas entradas correspondentes.

θ - limiar do elemento.

Considere uma rede com somente uma camada intermediária,

onde:

$x \in \mathbb{R}^{N_e}$ - conjunto aplicado a entrada da rede;

P - matriz de pesos das conexões entre a camada de entrada e a camada intermediária;

Q - matriz de pesos das conexões entre a camada intermediária e a camada de saída;

$z \in \mathbb{R}^{N_h}$ - conjunto de saída da camada intermediária;

$y \in \mathbb{R}^{N_s}$ - conjunto calculado na saída da rede;

$s \in \mathbb{R}^{N_s}$ - conjunto desejado na saída da rede;

N_e - número de elementos de entrada;

N_h - número de elementos da camada intermediária;

N_s - número de elementos da camada de saída.

$\theta_h \in \mathbb{R}^{N_h}$ - conjunto com os valores de limiar da camada intermediária;

$\theta_y \in \mathbb{R}^{N_s}$ - conjunto com os valores de limiar da camada de saída;

O treinamento da rede começa inicializando-se as matrizes de pesos com valores aleatórios. Esta inicialização poderá influir na convergência da rede (Najafi, 1989). Pode-se descrever o processo de aprendizagem pelos seguintes passos:

Passo 1- Inicializam-se as matrizes de pesos P e Q e os limiares θ_h e θ_y com valores aleatórios;

Passo 2- Calculam-se,

- saída da camada intermediária por, $z = f(Px - \theta_h)$ (20)

- saída da rede, $y = f(Qz - \theta_y)$ (21)

- erro da camada de saída (vide demonstração a seguir)

$$\delta_j = (s_j - y_j)y_j(1-y_j), \quad j=1, \dots, N_s \quad (22)$$

Passo 3- Nesta fase as matrizes de pesos são atualizadas, partindo-se da saída para a entrada. No caso de apenas uma camada intermediária, a sequência é a seguinte:

- calcula-se $\Delta Q_{ij} = \eta \delta_i z_j$, $i = 1, \dots, N_s$, $j = 1, \dots, N_h$ (23)

- atualiza-se $Q^{(p)} = Q^{(p-1)} + \Delta Q$ (24)

- calculam-se os valores de erro da camada intermediária (vide demonstração a seguir):

$$\delta h_j = z_j (1 - z_j) \sum_{k=1}^{N_s} \delta_k Q_{kj}, \quad j=1, \dots, N_h \quad (25)$$

- calcula-se $\Delta P_{ij} = \eta \delta h_i x_j$, $i=1, \dots, N_h$, $j = 1, \dots, N_e$ (26)

- atualiza-se $P^{(p)} = P^{(p-1)} + \Delta P$ (27)

O processo deve ser repetido com a introdução de um novo padrão e voltando-se ao passo 2.

Na seção seguinte, demonstra-se as expressões usadas nos passos anteriores.

5. Demonstração da Regra Delta Generalizada

Seja uma rede com somente uma camada intermediária e considere ζ_i a entrada líquida do elemento i , isto é, o produto escalar dos pesos de entrada da unidade pelos valores introduzidos:

$$\zeta_1 = \sum_{j=1}^{N_h} Q_{1j} z_j - \theta_1 \quad (28)$$

A saída do elemento após a aplicação da função de ativação f será:

$$y_1 = f(\zeta_1) \quad (29)$$

Para que o erro mínimo seja atingido a variação da matriz de pesos, à cada apresentação deve ser proporcional ao negativo do erro quadrático, isto é:

$$\Delta Q_{1j} \propto - \frac{\partial E^{(p)}}{\partial Q_{1j}} \quad (30)$$

onde $E^{(p)}$, corresponde ao valor do erro quadrático na saída da rede.

Se ΔQ for proporcional à derivada negativa do erro, em relação ao peso da conexão, garante-se que a cada nova apresentação, se aproxima do erro mínimo. Para se calcular a derivada pode-se aplicar:

$$\frac{\partial E^{(p)}}{\partial Q_{1j}} = \frac{\partial E^{(p)}}{\partial y_j} \frac{\partial y_j^{(p)}}{\partial Q_{1j}} \quad (31)$$

Da expressão (28) tem-se:

$$\frac{\partial \zeta_1}{\partial Q_{1j}} = \frac{\partial}{\partial Q_{1j}} (Q_{11} z_1 + \dots + Q_{1j} z_j + \dots + Q_{1, N_h} z_{N_h}) \quad (32)$$

Portanto,

$$\frac{\partial \zeta_1}{\partial Q_{1j}} = z_j \quad (33)$$

Seja,

$$\delta_1 = \partial E^{(p)} / \partial \zeta_1 \quad (34)$$

Da expressão (09) tem-se:

$$E^{(p)} = \frac{1}{2} ((s_1 - y_1)^2 + \dots + (s_1 - y_1)^2 + \dots + (s_{Ns} - y_{Ns})^2) \quad (35)$$

Sendo $y_1 = f(\zeta_1)$, tem-se:

$$\partial E^{(p)} / \partial \zeta_1 = (s_1 - y_1) f'(\zeta_1) \quad (36)$$

A equação (31) fica:

$$\partial E^{(p)} / \partial Q_{1j} = \delta_1 \cdot z_j \quad (37)$$

onde,

$$\delta_1 = \partial E^{(p)} / \partial \zeta_1 = (s_1 - y_1) f'(\zeta_1) \quad (38)$$

Adotando-se uma constante de proporcionalidade η , a equação (30) se torna:

$$\Delta Q_{1j} = -\eta \frac{\partial E^{(p)}}{\partial Q_{1j}} \quad (39)$$

e, substituindo-se (37) em (39):

$$\Delta Q_{1j} = -\eta \delta_1 z_j \quad (40)$$

Assim sendo, através de (40) se calcula o incremento a ser

adicionado a matriz Q a cada apresentação. Para o caso da matriz P, a mesma expressão não pode ser aplicada pois não existe uma saída desejada para a camada intermediária. Determina-se então o valor de incremento da matriz P capaz de minimizar o erro na saída da rede, isto é:

$$\Delta P_{uv} \propto - \partial E^{(p)} / \partial P_{uv} \quad (41)$$

Pela equação (41) impõe-se à matriz P uma variação ΔP_{uv} que provoca uma redução do erro quadrático, na saída da rede e na direção do gradiente. Assim sendo,

$$\frac{\partial E^{(p)}}{\partial P_{uv}} = \frac{\partial E^{(p)}}{\partial h_u} \frac{\partial h_u}{\partial P_{uv}} \quad (42)$$

onde h_u é a entrada líquida do elemento u da camada intermediária, isto é:

$$h_u = P_{u,1} x_1 + \dots + P_{u,v} x_v + \dots + P_{u,N_e} x_{N_e} \quad (43)$$

Portanto, $\frac{\partial h_u}{\partial P_{u,v}} = x_v$ (44)

Seja então, $\delta_{h_u} = \partial E^{(p)} / \partial h_u$ (45)

Substituindo-se (44) e (45) em (42), tem-se:

$$\partial E^{(p)} / \partial P_{u,v} = \delta_{h_u} x_v \quad (46)$$

e substituindo-se (46) em (41) tem-se:

Assim sendo, as derivadas da equação (51) ficam:

$$\left\{ \begin{array}{l} \frac{\partial y_1}{\partial z_u} = f'(\zeta_1) \cdot Q_{1u} \\ \dots \\ \frac{\partial y_{N_s}}{\partial z_u} = f'(\zeta_{N_s}) \cdot Q_{N_s u} \end{array} \right. \quad (55)$$

Substituindo-se (55) em (51) tem-se:

$$\frac{\partial E^{(p)}}{\partial z_u} = (s_1 - y_1) f'(\zeta_1) Q_{1u} + \dots + (s_{N_s} - y_{N_s}) \cdot f'(\zeta_{N_s}) \cdot Q_{N_s u} \quad (56)$$

e sendo os valores de δ da camada de saída,

$$\begin{array}{l} \delta_1 = (s_1 - y_1) f'(\zeta_1) \\ \dots \\ \delta_{N_s} = (s_{N_s} - y_{N_s}) f'(\zeta_{N_s}) \end{array} \quad (57)$$

a equação (51) fica,

$$\frac{\partial E^{(p)}}{\partial z_u} = \delta_1 Q_{1u} + \dots + \delta_{N_s} Q_{N_s u} \quad (58)$$

ou,

$$\frac{\partial E^{(p)}}{\partial z_u} = \sum_{k=1}^{N_s} \delta_k Q_{ku} \quad (59)$$

Substituindo-se (49) e (59) em (48), obtém-se a expressão de delta da camada intermediária:

$$\delta_{h_u} = f'(h_u) \sum_{k=1}^{N_s} \delta_k Q_{ku} \quad (60)$$

e substituindo-se (60 em (47) obtém-se:

$$\Delta P_{uv} \propto -x_v f'(h_u) \sum_{k=1}^{N_s} \delta_k Q_{ku} \quad (61)$$

Adotando-se uma constante de proporcionalidade η obtém-se a variação da matriz P:

$$\Delta P_{uv} = -\eta x_v f'(h_u) \sum_{k=1}^{N_s} \delta_k Q_{ku} \quad (62)$$

Através das expressões (40) e (62) pode-se atualizar as matrizes de peso Q e P respectivamente. Observa-se que para o cálculo dos valores de δ da camada intermediária usou-se o valor de δ da camada de saída. Este procedimento pode ser estendido para o caso de uma rede de várias camadas bastando para isso, utilizar-se sempre os valores de δ da camada posterior na expressão (62). Observa-se também que nas expressões (40) e (62) emprega-se a derivada da função f o que implica que a mesma seja diferenciável. Se for usada a função sigmóide, dada por:

$$y = \frac{1}{1 + e^{-(\alpha + \theta)}} \quad (63)$$

então a sua derivada em relação a entrada líquida α será:

$$\partial y / \partial \alpha = y(1-y) \quad (64)$$

Aplicando-se a expressão (64) na expressão (39), obtém a expressão de δ da camada de saída:

$$\delta_j = (s_j - y_j) y_j(1-y_j) \quad (65)$$

Como,
$$Q_{ij}^{(p+1)} = Q_{ij}^{(p)} + \Delta Q_{ij} \quad (66)$$

Obtém-se
$$Q_{ij}^{(p+1)} = Q_{ij}^{(p)} - \eta \delta_j^{(p)} z_i \quad (67)$$

Para atualizar a matriz P basta aplicar a expressão (64) à expressão (60):

$$\delta_{h_u} = z_u(1-z_u) \sum_{k=1}^{N_s} \delta_k Q_{ku} \quad (68)$$

De (62) tem-se
$$P_{uv}^{(p+1)} = P_{uv}^{(p)} - \eta x_v \delta_{h_u}^{(p)} \quad (69)$$

Através das expressões (65), (67), (68) e (69) pode-se realizar a etapa conhecida como propagação retroativa, onde as matrizes são atualizadas a cada apresentação. Este procedimento pode ser facilmente estendido para o cálculo de uma rede com mais de uma camada intermediária, bastando que se aplique as expressões (68) e (69), usando-se o valor de δ da camada posterior na expressão do somatório da expressão (68). ■

Observação 1: Para o caso de aproximação funcional, a camada de saída pode ser composta de elementos lineares, visto que a camada intermediária é suficiente para efetuar a aproximação funcional não linear. Com este procedimento se evita a necessidade de se normalizar a saída da rede, como no caso em que funções sigmóides são colocadas na última camada, forçando os valores de saída a ficarem limitados ao intervalo (0,1). ■

Observação 2: Considerando-se que diversos mapeamentos não lineares são compostos de uma parcela linear, este fato pode ser aproveitado e, com isso, facilitar em grande parte o treinamento da rede. Para tanto, basta estabelecer uma ligação direta entre a entrada e a saída da rede ajustando-se os pesos dessas conexões pela mesma regra do gradiente descendente. Esta técnica foi inicialmente introduzida em Rumelhart(1987). Em Wang(1989), foram analisadas as condições de treinabilidade de uma rede onde todos os elementos de uma camada são conectados aos elementos das camadas posteriores. Malakooti(1990), empregou esse tipo de rede na solução de problemas de otimização. ■

Observação 3: Para se estimar o número de elementos da camada intermediária deve-se considerar as condições de treinabilidade e a precisão desejada no mapeamento. Em geral aumentando-se o número de elementos na camada intermediária se obtém uma representação mais precisa. No Cap. I deste trabalho, foi introduzido um método prático para se estimar o número de elementos das camadas intermediárias baseado nas condições de treinabilidade da rede. ■

Na seção seguinte a rede de ligações funcionais será analisada.

$$\Delta P_{uv} \propto -\delta_{h_u} x_v \quad (47)$$

A dedução da expressão dos valores de δ da camada intermediária poderá ser feita subdividindo-se a expressão (45) em dois fatores, isto é:

$$\delta_{h_u} = \frac{\partial E^{(p)}}{\partial z_u} \frac{\partial z_u}{\partial h_u} \quad (48)$$

Observe o que cada uma dessas derivadas representa:

$\partial E^{(p)} / \partial z_u$ - reflete a variação do erro quadrático na saída da rede, em relação a saída do elemento u da camada intermediária (z_u);

$\partial z_u / \partial h_u$ - reflete a variação da saída do elemento u da camada intermediária em relação a entrada líquida (h_u) do mesmo elemento.

Determinação de $\partial z_u / \partial h_u$

Sendo f a função de ativação da unidade v , a saída da unidade será $z_u = f(h_u)$. Derivado-se:

$$\partial z_u / \partial h_u = f'(h_u) \quad (49)$$

Trata-se portanto da derivada da função de ativação f calculada sobre a entrada líquida da unidade da camada intermediária.

Determinação de $\partial E^{(p)} / \partial z_u$

Este é o fator que realmente diferencia uma unidade pertencente à camada intermediária de uma unidade da camada de saída.

Detalhando-se a expressão (35) tem-se:

$$E^{(p)} = \frac{1}{2} ((s_1 - f(\zeta_1))^2 + \dots + (s_1 - f(\zeta_1))^2 + \dots + (s_{N_s} - f(\zeta_{N_s}))^2) \quad (50)$$

Derivando-se o erro total na saída da rede, em relação a uma das saídas da camada intermediária, tem-se:

$$\frac{\partial E^{(p)}}{\partial z_u} = (s - y) \frac{\partial y_1}{\partial z_u} + \dots + (s_{N_s} - y_{N_s}) \frac{\partial y_{N_s}}{\partial z_u} \quad (51)$$

Considerando-se que

$$\begin{cases} y_1 = f(Q_{11} z_1 + \dots + Q_{1u} z_u + \dots + Q_{1N_h} z_{N_h}) \\ \dots \\ y_{N_s} = f(Q_{N_s 1} z_1 + \dots + Q_{N_s u} z_u + \dots + Q_{N_s N_h} z_{N_h}) \end{cases} \quad (52)$$

e sendo:

$$\begin{cases} \zeta_1 = Q_{11} z_1 + \dots + Q_{1u} z_u + \dots + Q_{1N_h} z_{N_h} \\ \dots \\ \zeta_{N_s} = Q_{N_s 1} z_1 + \dots + Q_{N_s u} z_u + \dots + Q_{N_s N_h} z_{N_h} \end{cases} \quad (53)$$

e também:

$$\begin{cases} y_1 = f(\zeta_1) \\ \dots \\ y_{N_s} = f(\zeta_{N_s}) \end{cases} \quad (54)$$

6. Rede Neural de ligações funcionais

A rede neural de ligações funcionais (Pao,1989) consiste basicamente em uma rede sem camadas intermediárias, porém com uma expansão não linear da entrada. Desta forma, a entrada da rede é ligada diretamente à saída através de uma rede linear e também através de funções escolhidas previamente. Em princípio qualquer função pode ser usada como expansão, tais como, funções trigonométricas ou expansões polinomiais. Este tipo de rede se aplica para os casos em que haja algum conhecimento das funções que constituem o mapeamento a ser estimado. Para os casos em que pouco, ou nenhum conhecimento for disponível, o uso da rede de várias camadas é aconselhável. Em Wang & Teixeira(1990) as condições de treinabilidade das redes neurais de ligações funcionais foram analisadas.

Observação 4: Uma opção bastante viável e que traz excelentes resultados no que se refere a treinabilidade da rede, é a combinação da Rede Neural de Várias Camadas com a Rede Neural de Ligações Funcionais. Esta combinação deve ser feita para os casos em que se conhecem algumas das funções que envolvem o mapeamento, o facilita sobremaneira o treinamento da rede. Uma vez estabelecidas as expansões funcionais, trata-se a entrada original mais a expansão funcional como sendo o conjunto de entrada da rede de várias camadas e aplica-se normalmente o algoritmo de propagação retroativa. ■

APÊNDICE II

Sistemas Não Lineares

Conceitos Básicos - Notação usada

Apresenta-se nesta seção os conceitos básicos referentes a teoria de sistemas dinâmicos não lineares, além de detalhar a notação usada no trabalho.

A seguinte bibliografia é sugerida: James(1987) para espaços topológicos, Boothby(1986) para teoria de manifolds diferenciáveis, Buck(1978) para noções de cálculo avançado e Isidori(1989) para teoria de sistemas dinâmicos não lineares.

1. Interior de um conjunto S: O Interior de um conjunto S é um subconjunto de S que consiste de todos os pontos que sejam internos a S.
2. Conjunto Aberto: Um conjunto S é considerado aberto se qualquer ponto de S for interno; neste caso, S coincide com o seu próprio Interior.
3. Exterior de um conjunto S: Um ponto p que não está no conjunto S estará no Exterior de S se todos os pontos suficientemente próximos de p também estiverem fora do conjunto S.
4. Complemento do conjunto S: O Complemento do conjunto S é o conjunto de pontos do espaço que não estão em S.
5. Conjunto Fechado: O conjunto S será fechado em \mathbb{R}^n se o complemento de S for um conjunto aberto.

6. Ponto Limite: Um Ponto Limite de um conjunto S é um ponto que não está nem no Interior nem no Exterior do conjunto S.

7. Fechamento de um conjunto S (\bar{S}): O Fechamento de um conjunto S é o conjunto formado pela união de S com todos os seus pontos limites, isto é:

$$\bar{S} = \text{Fechamento de } S = S \cup \text{limite}(S)$$

8. Sequência Crescente: Uma sequência $\{a_n\}$ será crescente se:

$$a_1 \leq a_2 \leq a_3 \leq \dots$$

9. Sequência Decrescente: Uma sequência $\{a_n\}$ será decrescente se:

$$a_1 \geq a_2 \geq a_3 \geq \dots$$

10. Sequência Monótona: Uma sequência será monótona se for crescente ou decrescente e será estritamente monótona se quaisquer dois termos consecutivos nunca forem iguais.

11. Notação de Funções: Seja A um subconjunto aberto de \mathbb{R}^n e considere a função $f:A \rightarrow \mathbb{R}$. O valor de f no ponto $x=(x_1, \dots, x_n)$ é representado por $f(x)=f(x_1, \dots, x_n)$.

12. Funções Suaves ou Funções da Classe C^∞ : A função f é considerada da Classe C^∞ ou simplesmente uma função suave se as suas derivadas parciais, de qualquer ordem, com relação a x_1, \dots, x_n , existirem e forem contínuas.

13. Funções Analíticas ou Funções da Classe C^ω : Uma função f é considerada da Classe C^ω ou simplesmente uma função Analítica se for suave e se, a cada ponto $x^\circ \in A$, existir uma vizinhança U de x° onde a expansão de f pela série de Taylor em x° , converge para $f(x)$, para qualquer $x \in U$.

14. Função Bijetiva (Bijeção): Uma função $f:A \rightarrow B$ é bijetiva se existir uma função $g:B \rightarrow A$ tal que $g \circ f$ seja igual a identidade em A e $f \circ g$ seja igual a identidade em B (\circ representa composição de funções).

15. Mapeamento: Um mapeamento $F:A \rightarrow \mathbb{R}^m$ é uma coleção (f_1, \dots, f_m) de funções $f_i:A \rightarrow \mathbb{R}$. Um mapeamento é C^∞ se todas as funções que o compõem forem C^∞ .

16. Matriz Jacobiana de F : A seguinte matriz corresponde a Matriz Jacobiana do mapeamento F no ponto x :

$$\frac{\partial F}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (01)$$

O valor de $\partial F/\partial x$ no ponto $x=x^\circ$ pode ser representada por $\left[\frac{\partial F}{\partial x}\right]_{x^\circ}$.

17. Teorema da Função Inversa (Vide demonstração em Buck, 1978)

Seja A um conjunto aberto de \mathbb{R}^n e seja $F:A \rightarrow \mathbb{R}^n$ um mapeamento C^∞ .

Se $\left[\frac{\partial F}{\partial x}\right]_{x^\circ}$ for não singular em um ponto $x^\circ \in A$,

então existirá uma vizinhança U de x° em A tal que $V = F(U)$ será

um conjunto aberto em \mathbb{R}^n e a restrição de F para U será um difeomorfismo sobre V (vide 34).

18. Teorema do Posto (Vide demonstração em Buck, 1978)

Considere os conjuntos abertos $A \subset \mathbb{R}^n$ e $B \subset \mathbb{R}^m$. Considere também que o mapeamento $F: A \rightarrow B$ seja C^∞ e que $[\partial F / \partial x]_x$ tenha ranque k para qualquer $x \in A$. Para cada ponto $x^0 \in A$ existirá uma vizinhança A_0 de x^0 em A , uma vizinhança B_0 de $F(x^0)$ em B , dois conjuntos abertos: $U \subset \mathbb{R}^n$ e $V \subset \mathbb{R}^m$, e dois difeomorfismos (vide 34): $G: U \rightarrow A_0$ e $H: B_0 \rightarrow V$, tal que $H \circ F \circ G(U) \subset V$, e tal que para qualquer $(x_1, \dots, x_n) \in U$:

$$H \circ F \circ G(x_1, \dots, x_n) = (x_1, \dots, x_k, 0, \dots, 0) \quad (02)$$

onde \circ representa composição de mapeamentos.

Observação 1: Considere o mapeamento $P_k: \mathbb{R}^n \rightarrow \mathbb{R}^m$ definido por:

$$P_k(x_1, \dots, x_n) = (x_1, \dots, x_k, 0, \dots, 0) \quad (03)$$

Então, desde que H e G sejam passíveis de inversão, pode-se determinar F a partir da expressão (02) por:

$$F = H^{-1} \circ P_k \circ G^{-1} \quad (04)$$

Esta expressão é válida para todos os pontos de A_0 .

19. Teorema da Função Implícita (Vide demonstração em Buck, 1978)

Sejam os conjuntos abertos $A \subset \mathbb{R}^m$ e $B \subset \mathbb{R}^n$ e o mapeamento C^∞ , $F: A \times B \rightarrow \mathbb{R}^n$. Considere que $(x, y) = (x_1, \dots, x_m, y_1, \dots, y_n)$ seja um ponto de $A \times B$.

Suponha que para algum ponto $(x^0, y^0) \in A \times B$, tenha-se:

$$F(x^0, y^0) = 0 \quad (05)$$

e que a matriz,

$$\frac{\partial F}{\partial y} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \dots & \frac{\partial f_1}{\partial y_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial y_1} & \dots & \frac{\partial f_n}{\partial y_n} \end{bmatrix} \quad (06)$$

seja não singular no ponto (x^0, y^0) . Haverá então uma vizinhança aberta A_0 de x^0 em A , uma vizinhança B_0 de y^0 em B e um mapeamento C^∞ único $G: A_0 \rightarrow B_0$ tal que:

$$F(x, G(x)) = 0 \quad (07)$$

para qualquer $x \in A_0$.

Corolário 1: Seja A um conjunto aberto em \mathbb{R}^n , seja M uma matriz $k \times n$ composta de funções C^∞ definidas em A e considere b um vetor de dimensão k composto de funções C^∞ definidas em A . Suponha que para um ponto $x^0 \in A$,

$$\text{ranque}(M(x^0)) = k. \quad (08)$$

Então, existirá uma vizinhança U de x^0 e um mapeamento C^∞ $G: U \rightarrow \mathbb{R}^n$ tal que

$$M(x)G(x) = b(x) \quad (09)$$

para qualquer $x \in U$.

Corolário 2: Seja A um conjunto aberto em \mathbb{R}^n , seja M uma matriz $k \times n$ composta de funções C^∞ definidas em A e considere b um vetor de dimensão k composto de funções C^∞ definidas em A . Suponha que para um ponto $x^\circ \in A$,

$$\text{ranque}(M(x^\circ)) = k. \quad (10)$$

Então, a equação

$$M(x)y = b(x) \quad (11)$$

terá, pelo menos, uma solução que é uma função C^∞ de x , em uma vizinhança de x° . Se $k = n$ esta solução será única.

20. Estrutura Topológica: Seja um conjunto S . Uma estrutura topológica em S é uma coleção de subconjuntos de S , denominados Conjuntos Abertos, que satisfazem as seguintes condições:

- a. A união de qualquer número de conjuntos abertos também é aberta;
- b. A interseção de um número finito de conjuntos abertos também é aberta;
- c. O conjunto S e o conjunto vazio \emptyset são conjuntos abertos.

21. Espaço Topológico: Denomina-se Espaço Topológico o espaço constituído pelo conjunto S em uma estrutura topológica.

22. Base de uma Estrutura Topológica: Uma base para uma estrutura topológica é constituída por uma coleção de conjuntos abertos,

denominados Conjuntos Abertos Básicos, com as seguintes propriedades:

a. S é a união de conjuntos abertos básicos;

b. uma interseção não vazia de dois conjuntos abertos básicos é uma união de conjuntos abertos básicos.

23. Vizinhança: Uma vizinhança de um ponto p de um espaço topológico pode ser qualquer espaço aberto que contenha p .

24. Noção de Continuidade: Sejam os espaços topológicos S_1 e S_2 e um mapeamento $F: S_1 \rightarrow S_2$. O mapeamento F será contínuo se a imagem inversa de cada conjunto aberto de S_2 for um conjunto aberto de S_1 .

25. Mapeamento Aberto: Um mapeamento $F: S_1 \rightarrow S_2$ é aberto se a imagem de um conjunto aberto de S_1 for um conjunto aberto de S_2 .

26. Homeomorfismo: Um mapeamento $F: S_1 \rightarrow S_2$ será um homeomorfismo se for bijetivo, contínuo e aberto.

27. Métrica: Uma métrica de um conjunto X é uma função não negativa $\rho: X \times X \rightarrow \mathbb{R}$ tal que:

a. $\rho(\xi, \xi) = 0$ para qualquer $\xi \in X$;

b. $\rho(\xi, \eta) = \rho(\eta, \xi)$ para qualquer $\xi, \eta \in X$;

c. $\rho(\xi, \zeta) \leq \rho(\xi, \eta) + \rho(\eta, \zeta)$ para qualquer $\xi, \eta, \zeta \in X$;

d. se $\rho(\xi, \eta) = 0$, para qualquer $\xi, \eta \in X$, então $\xi = \eta$.
para qualquer $\xi, \eta \in X$;

28. Métrica Euclidiana: A métrica euclidiana em um espaço real \mathbb{R}^n é definida por:

$$\rho((\xi_1, \dots, \xi_n), (\eta_1, \dots, \eta_n)) = ((\xi_1 - \eta_1)^2 + \dots + (\xi_n - \eta_n)^2)^{1/2} \quad (12)$$

29. Espaço Euclidiano: Um espaço Euclidiano Local X de dimensão n é um espaço topológico tal que, para cada ponto $p \in X$, exista um homeomorfismo ϕ mapeando alguma vizinhança de p em um conjunto aberto de \mathbb{R}^n .

30. Espaço Hausdorff: Considera-se que um espaço topológico S satisfaz o axioma de separação Hausdorff, (isto é: seja um espaço Hausdorff) se quaisquer dois pontos diferentes p_1 e p_2 tiverem vizinhanças disjuntas.

31. Manifold : Um espaço topológico será considerado um Manifold N de dimensão n se for localmente Euclidiano e Hausdorff e se tiver uma base contável.

32. Mapa de Coordenadas: Um mapa de coordenadas em um manifold N é um par (U, ϕ) , onde U é um conjunto aberto de N e ϕ é um homeomorfismo em \mathbb{R}^n .

33. Manifold Suave: Um manifold suave ou C^∞ é um manifold equipado com um mapa de coordenadas com funções suaves.

34. Difeomorfismo Considere que N e M sejam manifolds suaves de dimensão n . Um mapeamento $F: N \rightarrow M$ será um difeomorfismo se F for bijetivo e se F e F^{-1} forem mapeamentos suaves.

35. Campos Vetoriais: Campos vetoriais são mapeamentos suaves em relação aos seus argumentos, isto é, são constituídos de funções de valores reais com derivadas parciais contínuas de qualquer ordem.

36. Espaço Dual V^* : O espaço dual V^* de um espaço V é composto pelo conjunto de todas as funções lineares de valor real definidas em V .

37. Campos Covetoriais: Campos covetoriais são Objetos Duais, isto é, são mapeamentos suaves capazes de designar a cada ponto x (de um subconjunto U), o elemento do espaço dual $(\mathbb{R}^n)^*$.

Observação 2: Um espaço dual de um espaço vetorial de dimensão n é também um espaço vetorial cujos elementos são covetores.

38. Gradiente ou campo covetorial diferencial $d\lambda$: Seja λ uma função de valores reais definidos em um subconjunto aberto U de \mathbb{R}^n . Define-se gradiente ou campo covetorial por:

$$d\lambda(x) = \left[\frac{\partial \lambda}{\partial x_1} \quad \frac{\partial \lambda}{\partial x_2} \quad \dots \quad \frac{\partial \lambda}{\partial x_n} \right] \quad (13)$$

39. Derivada de λ ao longo de f : Seja λ uma função de valor real e f um campo vetorial. Suponha que ambos estejam definidos em um subconjunto U de \mathbb{R}^n . Define-se derivada de λ ao longo de f por:

$$L_f \lambda = (d\lambda(x), f(x)) = \frac{\partial \lambda}{\partial x} f(x) = \sum_{i=1}^n \frac{\partial \lambda}{\partial x_i} f_i(x) \quad (14)$$

Observação 3: Pode-se aplicar repetidamente a derivada L, isto é:

$$L_g L_f \lambda(x) = \frac{\partial(L_f \lambda)}{\partial x} g(x) \quad (15)$$

Observação 4: Se λ for diferenciada k vezes ao longo de f tem-se:

$$L_f^k \lambda(x) = \frac{\partial(L_f^{k-1} \lambda)}{\partial x} f(x) \quad (16)$$

onde $L_f^0 \lambda(x) = \lambda(x)$.

40. Produto de Lie ou Colchete de Lie: é definido por:

$$[f, g](x) = \frac{\partial g}{\partial x} f(x) - \frac{\partial f}{\partial x} g(x) \quad (17)$$

onde f e g são campos vetoriais em um conjunto aberto U de \mathbb{R}^n , e

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad \frac{\partial g}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_1}{\partial x_n} \\ \dots & & \dots \\ \frac{\partial g_n}{\partial x_1} & \dots & \frac{\partial g_n}{\partial x_n} \end{bmatrix} \quad (18)$$

Observação 5: A seguinte notação também é encontrada na literatura:

$$\text{ad}_f g(x) = [f, g](x) \quad (19)$$

Guez, A., Eilbert, J., Kam, M. "Neural Network Architecture for Control", IEEE Control System Magazine, April 1988.

Haaser, N. & Sullivan, J. "Real Analysis", New York: Van Nostrand Reinhold, 1971.

Hermes, H. "Involutive Distributions and Canonical Forms For Distributions and Control Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holand), 1986.

Hirschorn, R. "Invertibility of Multivariable Nonlinear Control Systems", IEEE Transactions on Automatic Control, Vol. AC-24, No.6, December 1979.

Hirschorn, R. "Output Tracking in Multivariable Nonlinear Systems", IEEE Transactions on Automatic Control, Vol. AC-26, No.2, April 1981.

Hopfield, J. "Learning Algorithms and Probability Distributions in Feed-Forward and Feed-Back Networks", Proc. Natl. Acad. Sci. USA., Vo.84, pp.8429-8433, December 1987.

Hornik, K., Stinchcombe, M., White, H. "Multi-layer Feedforward Networks are Universal Aproximators", Neural Networks, Vol.2, 1989.

Hornik, K., Stinchcombe, M., White, H. "Universal Approximation of an Unknwon Mapping and Its Derivatives Using Multilayer Feedforward Networks", Neural Networks, Vol.3, pp. 551-560, 1990.

Hunt, R., Su, R. Meyer, G. Global Transformation of Nonlinear Systems, " IEEE Trans. Aut. Control, AC-28, No.1, pp 24-31, 1983.

Isidori, A., et al, "Nonlinear Decoupling via Feedback: A Differential Geometric Approach", IEEE Transactions on Automatic Control, Vol. AC-26, pp.331-345, 1981.

Isidori, A "Nonlinear Control Systems", Springer-Verlag, 2nd Edition, 1989.

Isidori, A. & Byrnes, C. "Output Regulation of Nonlinear Systems"; IEEE Transactions on Automatic Control, Vol. 35, N.2, Feb, 1990.

Ivakhnenko, A. "Heuristic Self-Organization in Problems of Engineering Cybernetics", Automatica, Vol.6, pp.207-219, Pergamon Press, 1970.

Ivakhnenko, A. & Koppa, Y. "Regularization of Decision Functions in the Group Method of Data Handling", Soviet Automatic Control, Vol. 15, No. 2, 1970.

James, I. "Topological and Uniform Spaces", Springer-Verlag, 1987.

Jacob, H. "The Concept of Rationality and Stabilization of Nonlinear Systems"; Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holand), 1986.

Jakubczyk, B. "Feedback Linearization of Discrete Time Systems", Systems And Control Letters, No. 9, pp 411-416, 1987a.

43. Ponto Regular de uma Distribuição: Diz-se que um ponto x^0 de U é um ponto regular de uma distribuição Δ , se existir uma vizinhança U^0 de x^0 com a propriedade de que Δ seja não singular em U^0 . Cada ponto de U que não for um ponto regular é conhecido como ponto de singularidade.

44. Distribuição Involutiva: Uma distribuição Δ é involutiva se o Produto Lie $[\tau_1, \tau_2]$ de qualquer par de campos vetoriais τ_1, τ_2 pertencentes a Δ for também um campo vetorial que pertence a Δ , isto é, se:

$$\tau_1 \in \Delta, \tau_2 \in \Delta \rightarrow [\tau_1, \tau_2] \in \Delta \quad (24)$$

Observação 6: Pode-se aplicar a operação de Produto Lie repetidamente, isto é:

$$\text{ad}_f^k g(x) = [f, \text{ad}_f^{k-1} g(x)](x) \quad (20)$$

para qualquer $k \geq 1$, e sendo $\text{ad}_f^0 g(x) = g(x)$.

41. Distribuições: Considere que os campos vetoriais f_1, \dots, f_d sejam definidos em um conjunto aberto U . Considere que em qualquer ponto fixo x em U , os vetores $f_1(x), \dots, f_d(x)$ sejam base para o espaço vetorial $\Delta(x)$ no qual todos os $f_i(x)$ estão definidos, isto é:

$$\Delta(x) = \text{span}\{f_1(x), \dots, f_d(x)\} \quad (21)$$

Desta forma para cada ponto x do conjunto U , designou-se um espaço vetorial. A designação $\Delta(x)$ é chamada Distribuição. A seguinte notação é também usada para definir a designação total:

$$\Delta = \text{span}\{f_1, \dots, f_d\}, \quad (22)$$

sendo $\Delta(x)$ o espaço definido por Δ no ponto x .

42. Distribuição Não Singular: Uma distribuição Δ é não singular se existir um inteiro d tal que:

$$\dim(\Delta(x)) = d \quad (23)$$

para qualquer x em U .

Observação 7: Uma distribuição singular pode ser também denominada como uma distribuição de dimensões variáveis.

Jakubczyk, B., & Sontag, E. "The Effect of Sampling on Feedback Linearization", Proceedings of the 26th Conference on Decision And Control, Los Angeles, CA, December 1987b.

Josin, G. "Integrating Neural Networks with Robotics", AI Expert, August 1988.

Kolmogorov, A., "On The Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition. Dokl Akad Navk USSR, Vol.14, pp.953-956, 1957.

Kohonen, T. "Self-Organization and Associative Memory", Springer-Verlag, New York, 1984.

Karnin, E. "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks", IEEE Transactions on Neural Networks, Vol.1, No.2, June 1990.

Kawato, M., Uno, Y., Isobe, M., Suzuki, R. "Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics", IEEE Control System Magazine, April, 1988.

Kraft, L. & Campanha, D. "A Comparison Between Cmac Neural Network Control and Two Traditional Adaptive Control Systems", IEEE Control Systems Magazine, April 1990.

Krener, A. "On The Equivalence of Control Systems and The Linearization of Nonlinear Systems", SIAM J. Control Optim., 11, No. 4, pp 670-676, 1973.

Kuo, B. "Automatic Control Systems, Englewood Cliffs, New Jersey, Prentice Hall Inc., 1970.

Landau, I., Cyrot, D., Montano, A. "Adaptive Control of a Class of Nonlinear Discrete Time systems - Application to a Heat Exchanger", Proceedings of the 26th Conference on Decision and Control, Los Angeles, CA, December 1987.

Lee, H. & Marcus, S. "Approximate and Local Linearization of Nonlinear Discrete Time Systems", Int. J. Control, 44, pp 1103-1124, 1986.

Lee, C. & Maxwell, T. "Learning, Invariance, and Generalization in High-order Neural Networks", Applied Optics, Vol. 26, N. 23, December 1987a.

Lee, H. & Marcus, S. "On the Immersion of a Discrete Time Nonlinear System into a Linear System", Proceedings of the 26th Conference on Decision and Control, Los Angeles, CA, December 1987b.

Lee, H., Apostolakis, A., Marcus, S. "Linearization of Discrete Time Nonlinear Systems", Proc. Am. Control Conf., Seattle, pp. 853-862, 1987c.

Lee, H., Apostolakis, A., Marcus, S. "Linearization of Discrete Time Nonlinear Systems", Int. J. Control, 45, pp 1803-1822, 1987d.

Lee, H., Apostolakis, A., Marcus, S. "Remarks on Discretization and Linear Equivalence of Continuous Time Nonlinear Systems", 26th IEEE Conf. on Decision and Control, pp 1783-1785, 1987e.

Bibliografia

Aeyels, D. "Local and Global Stabilizability for Nonlinear Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V(North-Holand), 1986.

Akhrif, O. "Nonlinear Adaptive Control with Application to Flexible Structures", Ph.D. Thesis, University of Maryland, 1989.

Antsaklis, P. "Neural Networks for Control Systems", IEEE Transactions on Neural Networks, Vol.1, No.2, June 1990.

Aström, K. & Wittenmark, B. "Computer Controlled Systems, Englewood Cliffs, New Jersey, Prentice Hall Inc., 1984.

Atlas, L. et al. "A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees", Proceedings of the IEEE, Vol.78, No.10, October 1990.

Barhen, J., Gulati, S., Zak, M. "Neural Learning of Constrained Nonlinear Transformations", IEEE Computer Magazine, June 1989.

Barto, A., Anderson, C. & Sutton, R. "Synthesis of Nonlinear Surfaces by a Layered Associative Search Network", Biological Cybernetics, Springer-Verlag, 1982.

Baumann, W. & Rugh, W. "Feedback Control of Nonlinear Systems by Extended Linearization: The Multi-input Case", Theory and Application of Nonlinear Control Systems, C.J.Byrnes and A. Lindquist (editors), Elsevier Science Publishers B. V. (North-Holland), 1986.

Bavarian, B. "Introduction to Neural Networks for Intelligent Control", IEEE Control Systems Magazine, April 1988.

Bishop, R. & Goldberg, S. "Tensor Analysis on Manifolds", Dover Publications, Inc., New York, 1980.

Boothby, W. "An Introduction to Differentiable Manifolds and Riemannian Geometry", Academic Press, 2nd Edition, 1986.

Brockett, R. "Volterra Series and Geometric Control Theory", Automatica, Vol. 12, pp. 167-176, Pergamon Press, 1976.

Buck, R. "Advanced Calculus", McGraw Hill, 3rd Edition, 1978.

Chen, F. "Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control", IEEE Control Systems Magazine, April, 1990.

Chizeck, H. "Controller Design for Neuroprosthetic Systems", IEEE International Conference on Control Applications, Jerusalem, pp. 1.1-RP3-1,7, (1989)

Chu, S., Shoureshi, R., Tenório, M. "Neural Networks for Systems Identification", IEEE Control Systems Magazine, April 1990.

Chung, S. & Grizzle, J. "Observer Error Linearization for Sampled-Data Systems", Proceedings of the 28th Conference on Decision and Control, Tampa, Florida, December 1989.

Cotter, N. "The Stone-Weierstrass Theorem and Its Application to Neural Networks", IEEE Transactions on Neural Networks, Vol. 1. N.4, December 1990.

Culhane, A. & Peckerar, M. "A Neural Net Approach to Discrete Hartley and Fourier Transforms", IEEE Transactions on Circuits and Systems, Vol. 36, NO. 5, May 1989.

El-Bially, A. "Control of Multiplicative Discrete-Time Systems, Ph.D. Thesis, Case Western Reserve University, Systems Engineering Department, 1990.

Fliess, M "A Remark on Nonlinear Observability", IEEE Transactions On Automatic Control, Vol. AC-27, No.2, April 1982.

Fliess, M "Some Remarks on Nonlinear Invertibility and Dynamic State-Feedback", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V. (North-Holand), 1986a.

Fliess, M "A Note on the Invertibility of Nonlinear Input-Output Differential Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V. (North-Holand), 1986b.

Fogel, D., Fogel, L., Porto, V. "Evolving Neural Networks", Biological Cybernetics, Springer-Verlag 1990.

Freund, E. "The Structure of Decoupled Nonlinear Systems,", Int. J. Contr., Vol. 21, pp. 651-654, 1975.

Funahashi, K. "On the Approximate Realization of Continuous Mappings by Neural Networks"; Neural Networks, Vol. 2, pp. 183-192, 1989.

Geng, K. "Real-time Parameter Identification of a class of Nonlinear Discrete-time Models of Electrically Stimulated Muscle, MS. Thesis, Case Western Reserve University, 1989.

Girosi, F. & Poggio, T. "Networks and the Best Approximation Property", Biological Cybernetics, Springer-Verlag 1990.

Grizzle, J. "On a Geometric Approach for Discrete Time Decoupling Problems", Proceedings of 24th Conference on Decision and Control, pp.366-370, 1985a.

Grizzle, J. "Controlled Invariance For Discrete Time Nonlinear Systems", IEEE Trans. Aut. Control. Vol.AC-30, No.9, pp.868-874, 1985b.

Grizzle, J. "Local Input-Output Decoupling of Discrete Time Nonlinear Systems", Int. J. Control, Vol. 43, No. 5, pp 1517-1530, 1986a.

Grizzle, J. "Feedback Linearization of Discrete Time Systems", Lectures Notes in Control and Information Science, Vo.83, Springer, Berlin-New York, pp 273-281, 1986b.

Grizzle, J. "On Infinite Zeros of Nonlinear Discrete-time Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V(North-Holand), 1986c.

Grizzle, J. & Kokotovic, P. "Feedback Linearization of Sampled-Data Systems", IEEE Trans. Aut. Control, Vo. 33, No. 9, pp 857-859, 1988.

Lee, H. & Marcus, S. "On Input-Output Linearization of Discrete Time Nonlinear Systems", Systems and Control Letters, No. 8, pp 249-259, 1987f.

Lee, H., Aparostaltis, A., Marcus, S. "Linearization of Discrete and Discretized Nonlinear Systems", in Lecture Notes in Control and Information Science, pp 91-101, 1988a.

Lee, H., Aparostaltis, A., Marcus, S. "On the Digital Control of Nonlinear Systems", 27th Conf. On Decision and Control, pp 480-481, 1988b.

Lee, H. & Aravena, J. "Digital Control of a Robot Manipulator Using Nonlinear Feedback", Proceedings of the 28th Conference on Decision and Control, Tampa, Florida, December 1989.

Li, W. & Slotine, J. "Neural Network Control of Unknown Nonlinear Systems", American Control Conference, Pittsburgh, 1989.

Lippmann, R. "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, pp.4-22, April 1987.

Loparo, K. & Teixeira, E. "A New Approach for Adaptive Control of Nonlinear Systems Using Neural Networks", 1990 IEEE International Conference On Systems Man and Cybernetics, Los Angeles, CA, November 1990.

Marino, R. "Hamiltonian Techniques in Control of Robot Arms and Power Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holland), 1986.

Meyer, G., Cicolani, L. "Applications of Nonlinear System Inverses to Automatic Flight Control Design-System Concepts and Flight Evaluation", Theory and Applications of Optional Control in Aerospace Systems, P. Kent. Ed. Nato, 1980.

Minsky, M. & Papert, S. "Perceptrons", MIT Press, Cambridge, Ma, 1969.

Monaco, S. & Norman-Cyrot, D. "The Immersion Under Feedback of Multidimensional Discrete Time Nonlinear System Into a Linear System", Int. J. Control, 38, pp 245-261, 1983a.

Monaco, S. & Norman-Cyrot, D. "Formal Power Series and Immersion Direct Under Feedback of Nonlinear Discrete Time System", 22nd IEEE Conf. Dec. and Control, pp. 655-660, 1983b.

Monaco, S. & Norman-Cyrot, D. "Invariant Distributions for Discrete Time Nonlinear Systems", Systems and Control Letters, 5, pp 191-196, 1984a.

Monaco, S. & Norman-Cyrot, D. "On the Realization of Nonlinear Discrete Time Systems", Systems and Control Letters, 5, pp 145-152, 1984b.

Monaco, S. & Norman-Cyrot, D. "Partial Realization of a Nonlinear Discrete-Time System from an Equilibrium Point", Proceedings of 23rd Conference on Decision and Control, Las Vegas, NV, December 1984c.

Monaco, S. & Norman-Cyrot, D. "On the Sampling of a Linear Analytic Control System", 24th IEEE Conf. On Decision and Control", pp 1457-1462, 1985.

Monaco, S. & Norman-Cyrot, D. "A Lie Exponential Formula For the Nonlinear Discrete Time Functional Expansions", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holand), 1986a.

Monaco, S. & Norman-Cyrot, D. "Nonlinear Systems in Discrete Time", in Algebraic and Geometric Methods in Nonlinear Control Theory, Ed. M. Fliess and M. Hazewinded, 1986b.

Monaco, S. & Norman-Cyrot, D. "Invariant Distribution Under Sampling", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holand), 1986c.

Monaco, S., Norman-Cyrot, D., Stornelli, S. "On the Linearizing Feedback in Nonlinear Sampled Data Control Schemes", 25th Conf. on Decision and Control, pp. 2056-2060, 1986d.

Monaco, S. & Norman-Cyrot, D. "Minimum-phase Nonlinear Discrete Time and Feedback Satabilization", 26th IEEE Conf. on Decision and Control, pp 979-986, 1987.

Monaco, S. & Norman-Cyrot, D. "Multirate Sampling and Nonlinear Digital Control", Submitted to Automatica, 1988a.

Monaco, S. & Norman-Cyrot, D. "Zero Dynamics of Sampled Nonlinear Systems", Systems and Control Letters, 10, 1988b.

Montano, A., Landau, L., Normand, D. "Adapative Control of Nonlinear Systems Using Nonlinear And Linear Models", Proceedings of the 27th Conference on Decision and Control, Austin Texas, December 1988.

Najafi, H., Nasiruddin, M., Samad, T. "Effect of Initial Weights on Back-Propagation and Its Variations", 1989 IEEE International Conference on Systems, Man, and Cybernetics, Cambridge, Ma, 1989.

Narendra, K. & Parthasarathy, K. "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, Vol.1, N.1, March 1990.

Nguyen, D. & Widrow, B. "Neural Networks for Self-Learning Control Systems", IEEE Control Magazine Systems, April 1990.

Pao, Y., "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley Publishing Company, Inc., 1989.

Porter, W. "Diagonalization and Inverses for Nonlinear Systems", Int. J. Contr., Vol.21, pp. 67-76, 1970.

Psaltis, D., Sideris, A., Yamamura, A. "A Multilayered Neural Network Controller", IEEE Control Systems Magazine, pp.17-21, April 1988.

Rebhuhn, D. "Invertibility of C^∞ Multivariable Input-Output Systems", IEEE Transactions On Automatic Control, Vol.AC-25, No.2, April 1980.

Respondek, W. "Partial Linearization, Decompositions and Fibre Linear Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holand), 1986.

Rumelhart, D., McClelland, J. "Parallel Distributed Processing", Vol.1, 2, MIT Press, Cambridge, MA, 4th Edition, 1987.

Teixeira, E. et al, "Feedback Control of Nonlinear Multiplicative Systems Using Neural Networks: An Application to Electrically Stimulated Muscle", IEEE Conference on Systems Engineering, August 1991b.

Tollenaere, T. "SuperSAB: Fast Adaptive Back Propagation with Good Scaling Properties", Neural Networks, Vol. 3, pp. 561-573, 1990.

Wang, J. & Malakooti, B. "On Training of Artificial Neural Networks", Proceedings of Int. Joint Conference on Neural Networks, Washington DC, June 1989.

Wang, J. & Teixeira, E. "On Trainability Conditions of the Functional Link Neural Network"; IEEE International Conference In Systems Engineering, Pittsburgh, August 1990.

Willsky, A. "On the Invertibility of Linear Systems", IEEE Transactions on Automatic Control, June 1974.

Report-1:

Teixeira, E., Loparo, K., Gomide, F. "Design of Multilayer Neural Networks for Accurate Identification of Nonlinear Mappings", Technical Report, Case Western Reserve University - Systems Engineering Department, 1991.

Sain, M., Massey, J. "Invertibility of Linear Time-Invariant Dynamical Systems", IEEE Transactions on Automatic Control, Vol.AC-14, No.2, April 1969.

Sastry, S. & Isidori, A.; "Adaptive Control of Linearizable Systems", IEEE Trans. on Autom. Control, Vol.34, No.11, pp.1123-1131, November 1989.

Sejnowski, T., Koch, C., Churchland, P. "Computational Neuroscience", Science, Vol.241, September 1988.

Shue, G. "Multiplicative Model of Stimulated Muscles for FNS Control", M.S. Thesis, Case Western Reserve University, 1990.

Silverman, L. "Inversion of Multivariable Linear Systems", IEEE Transactions on Automatic Control, Vol.AC-14, No.3, June 1969.

Singh, R. & Johnson, T. "A Functional Expansion Approach to the solution of Nonlinear Feedback Problems", IEEE Transactions on Automatic Control, Vol. AC-26, N.2, April 1991.

Singh, S. & Rugh, W. "Decoupling in a Class of Nonlinear Systems by State Variable Feedback", Transactions ASME J. Dyn. System Measur. Control, Vol.21, pp.651-654, 1975.

Singh, S. "A Modified Algorithm for Invertibility in Nonlinear Systems", IEEE Transactions on Automatic Control, Vol.AC-26, No.2, April 1981.

Singh, S. "Invertibility of Observable Multivariable Nonlinear Systems", IEEE Transactions on Automatic Control, Vol.AC-27, No.2, April 1982.

Spong, M. "Robust Stabilization for a Class of Nonlinear Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holand), 1986.

Stefani, G. "On the Local Controllability of a Scalar-Input Control System"; Systems", Theory and Application of Nonlinear Control Systems, Elsevier Science Publishers B.V (North-Holand), 1986.

Su, R. "On the Linear Equivalents of Nonlinear Systems", Systems and Control Letters, 2, No. 1, pp 48-52, 1982;

Schwartz, C. & Dickinson, B. "Some Finite Dimensional Realization Theory for Nonlinear Systems", Theory and Application of Nonlinear Control Systems, Elsevier S. Publishers B.V (North-Holand), 1986.

Taylor, D., Kokotovic, P., Marino, R. Kanellakopoulos, I. "Adaptive Regulation of Nonlinear Systems with Unmodeled Dynamics", IEEE Transactions on Automatic Control, Vol. 34, No. 4, April 1989.

Teixeira, E., Gomide, F., Andrade, M. "Extreme Conditions of Convergence for the Hopfield Neural Network", 1989 IEEE International Conference On Systems Man and Cybernetics, Cambridge, Ma, November 1989.

Teixeira, E., Loparo, K., Gomide, F.; "Design of Multi-Layer Neural Networks for Accurate Identification of Nonlinear Mappings", American Control Conference, Boston, June 1991a.