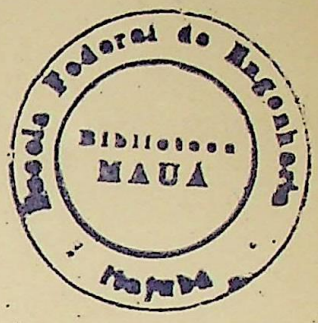
.54
94

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

DISSERTAÇÃO DE MESTRADO



"APLICAÇÃO DE REDES NEURONAIS EM CONTROLE DE PROCESSOS"

ERNESTO ALFONSO CASTILLO SATURNO

ORIENTADOR : PROF. GERMANO LAMBERT TORRES

CO-ORIENTADOR: PROF. LUIZ EDUARDO BORGES DA SILVA

ITAJUBÁ - MG

1992

CLASS.
CUTT. 5254a
TOMBO. 694

37-41

42-47

À minha família e amigos.

AGRADECIMENTOS

Este trabalho, só foi possível graças ao apoio e colaboração das seguintes pessoas:

Os meus pais, pelo incentivo e a confiança que em mim depositaram.

Aos professores, Germano Lambert Torres e Luiz Eduardo Borges da Silva, pela orientação e dedicação.

Aos colegas do GIA, pela amizade e o apoio nos momentos mais difíceis.

À CAPES, pelo apoio financeiro.

A todos os amigos da EFEI, que de uma forma ou outra ajudaram na realização deste.

RESUMO

Este trabalho tem como objetivo descrever a utilização de Redes Neuronais em Controle de Processos. Mais especificamente, estudamos a aplicação das redes nas funções de identificação e controle.

O sistema desenvolvido é aplicado ao controle de velocidade de um motor de corrente contínua.

A técnica proposta utiliza a capacidade de aprendizagem das redes neuronais na implementação de um esquema de controle adaptativo. Tal capacidade, permite à rede ser treinada para representar o comportamento dinâmico de um determinado processo, neste caso, o acionamento tiristorizado de um motor CC. A partir do modelo identificado, uma outra rede é treinada no papel de controlador, de tal forma que, a saída do processo acompanhe o sinal de referência da malha.

É apresentado também, um esquema de adaptação para funcionamento em regime de operação normal, cujo desempenho é verificado através de testes de variação da referência, e de situações inesperadas, como são, a variação de parâmetros físicos e a presença de ruído na realimentação.

Os resultados obtidos mostram a aplicabilidade do sistema apresentado, principalmente, em processos onde o nível de ruído é elevado, e outros tipos de controle resultam insatisfatórios.

ABSTRACT

This work describes the use of Neural Networks in Process Control. Specifically, we study its application in process identification and control.

The developed system is applied to a DC Motor speed control.

The proposed technique makes use of the learning capability of neural networks to implement an adaptive scheme. Such capability allows the network to learn the dynamic behavior of a given process. In our case, the SCR-driven DC motor is the process. This identification network is then used to train another network as the process controller, i.e., the process output follows the reference signal.

An adaptation scheme for working conditions is also presented, which performance is verified through testing of reference variations, and for unexpected situations, such as physical parameter variations and noise presence.

The results show the applicability of the neural control system, especially, in processes with high level of noise, where other control techniques are unsound.

CAPÍTULO 2 - CONCEITOS GERAIS SOBRE REDES NEURONAIS

2.1 - HISTÓRICO..... 37
2.1.1 - Antecedentes..... 37
2.1.2 - Queda do interesse..... 39
2.1.3 - Novo auge..... 41

2.2 - DESCRIÇÃO..... 42

2.3 - REDE MULTICAMADAS..... 44

2.4 - ALGORITMO DE TREINAMENTO "BACK-PROPAGATION"..... 45
2.4.1 - Função de ativação..... 46
2.4.2 - Erro parcial e erro global..... 47
2.4.3 - Desenvolvimento matemático..... 48
2.4.4 - Normalização..... 51
2.4.5 - Passos de treinamento..... 52
2.4.6 - Deficiências..... 54

2.5 - ÁREAS DE ATUAÇÃO..... 55

2.6 - IMPLEMENTAÇÃO PRÁTICA..... 56
2.6.1 - Softwares aplicativos..... 56
2.6.2 - Implementações em hardware..... 57

CAPÍTULO 3 - REDES NEURONAIS EM SISTEMAS DE CONTROLE

3.1 - REDES NEURONAIS NA MALHA DE CONTROLE..... 59

3.2 - IDENTIFICAÇÃO DE PROCESSOS..... 60

3.3 - O CONTROLADOR NEURONAL..... 64
3.3.1 - Controle Indireto..... 65
3.3.2 - Controle Direto..... 70

3.4 - ESQUEMA DE CONTROLE EM REGIME DE OPERAÇÃO NORMAL..... 71

SUMÁRIO

Agradecimentos..... i
Resumo..... ii
Abstract..... iii
Sumário..... iv

INTRODUÇÃO..... 1

CAPÍTULO 1 - SISTEMAS DE CONTROLE

1.1 - METODOLOGIAS DE CONTROLE..... 4
1.1.0 - Introdução..... 4
1.1.1 - Tipos de sistemas de controle..... 4
1.1.2 - Desempenho do sistema..... 7
1.1.2.1 - Sinais de teste..... 7
1.1.2.2 - Resposta transitória..... 8
1.1.3 - Regulador PID..... 9
1.1.4 - Controle Adaptativo..... 11
1.1.4.1 - Estrutura básica..... 12
1.1.4.2 - Desvantagens..... 15

1.2 - DESCRIÇÃO DO PROCESSO..... 16
1.2.0 - Introdução..... 16
1.2.1 - Conversor tiristorizado..... 19
1.2.1.1 - Escolha do conversor..... 21
1.2.1.2 - Ponte trifásica tiristorizada..... 22
1.2.1.3 - Tensão de saída do conversor..... 24
1.2.2 - Circuito de disparo..... 25
1.2.2.1 - Filosofias de disparo..... 26
1.2.2.2 - Circuito tipo cossenoidal..... 27
1.2.3 - Motor de corrente contínua..... 29
1.2.3.1 - Funcionamento do motor..... 29
1.2.3.2 - Modelagem..... 32
1.2.4 - Função de transferência do acionamento..... 35

3.5 - METODOLOGIA DE DESENVOLVIMENTO DE APLICAÇÕES..... 76
3.5.1 - Definição do problema..... 76
3.5.2 - Projeto..... 78
3.5.2.1 - Características dos neurônios..... 78
3.5.2.2 - Estrutura e tamanho das redes..... 78
3.5.2.3 - Parâmetros de treinamento..... 81
3.5.2.4 - Conjunto de padrões para treinamento.. 82
3.5.3 - Implementação..... 82
3.5.4 - Manutenção..... 83

CAPÍTULO 4 - SIMULAÇÃO DO CONTROLE DE VELOCIDADE UTILIZANDO REDES NEURONAIS

4.0 - INTRODUÇÃO..... 85

4.1 - SIMULAÇÃO DO ACIONAMENTO DO MOTOR CC..... 87
4.1.1 - Transformada Z..... 88
4.1.2 - Equação as diferenças..... 89
4.1.3 - Zero-order hold (ZOH)..... 90
4.1.4 - Escolha do período de amostragem..... 92
4.1.5 - Equação as diferenças do acionamento..... 93

4.2 - PROCEDIMENTO DE IDENTIFICAÇÃO - TESTES E RESULTADOS... 99

4.3 - TREINAMENTO DO CONTROLADOR - TESTES E RESULTADOS..... 124

4.4 - ANÁLISE DE DESEMPENHO..... 144
4.4.1 - Mudanças na referência..... 146
4.4.2 - Variação de parâmetros físicos..... 148
4.4.3 - Presença de ruído..... 151

CONCLUSÕES..... 154

BIBLIOGRAFIA..... 158

APÊNDICE A - Transformada Z..... 164

APÊNDICE B - Software NEUWORK..... 166

INTRODUÇÃO

Os sistemas de controle estão presentes em quase todas as áreas da ciência moderna, na engenharia, na medicina, nas telecomunicações, e muitas outras. Pode-se notar que a cada dia, tornam-se cada vez mais rigorosos os requisitos exigidos em termos de confiabilidade e fidelidade de resposta. Este fato incentiva a procura por soluções cada vez melhores e mais abrangentes para o controle de um dado processo. O surgimento de técnicas de controle moderno representam um caminho aberto para novos desenvolvimentos. No entanto, muito ainda deve ser feito para colocar em prática controles que operam sistemas com procedimentos complexos.

Neste trabalho pretendemos estudar o comportamento das Redes Neuronais em sistemas de controle, e como caso específico, no Controle de Velocidade de um Acionamento de Motor de Corrente Contínua. As Redes Neuronais são estruturas de processamento paralelo que imitam o funcionamento do cérebro humano, tentando incorporar funções como aprendizagem, memorização e generalização. As redes são constituídas por uma grande quantidade de processadores simples chamados Neurônios, interligados entre si de acordo com um determinado tipo de configuração.

Vários trabalhos tem sido publicados dentro desta linha de pesquisa, no entanto, aplicações dirigidas como a que pretendemos aqui ainda são muito poucas. Muitas são as dúvidas sobre o comportamento desta estrutura, e o surgimento de uma teoria básica sólida ainda demorará um pouco. As conclusões, na maioria dos casos, são muito específicas, o que impede a expansão das áreas de atuação. A maioria dos estudos conhecidos limitam-se ao treinamento das redes e seu teste sob condições normais. Faltam ainda, regras que garantam a permanência do desempenho registrado mesmo diante de situações inesperadas.

O objetivo deste trabalho é o de realizar estudos, através de simulações, do sistema de controle do processo escolhido funcionando junto com as redes neuronais, constituindo um sistema de controle do tipo adaptativo. Será verificado o comportamento e desempenho do sistema, tanto durante a fase de treinamento das redes, como durante o regime de funcionamento normal.

O desenvolvimento deste trabalho será dividido em quatro capítulos. No primeiro, será realizada uma pequena abordagem sobre algumas estruturas clássicas de controle e sobre a estrutura e funcionamento do sistema de controle adaptativo tradicional, sobre quais os aspectos que ainda representam um problema de engenharia e que podem justificar uma solução através de redes neuronais. Será descrito, em termos de diagrama de blocos e equações para simulação, o funcionamento de um acionamento de motor de corrente contínua.

No segundo capítulo, será apresentada uma base teórica sobre as redes neuronais, começando com um breve histórico e a situação atual das aplicações, seguindo com a descrição da rede do tipo multicamadas, que tem mostrado maior aplicabilidade em controle, e do algoritmo de treinamento *back-propagation*, o mais utilizado para este tipo de topologia. O capítulo termina com as principais áreas de atuação onde as redes tem mostrado um sucesso apreciável e uma noção sobre a implementação prática do sistema.

Já, no terceiro capítulo, o estudo será concentrado nas aplicações em controle, mais especificamente, na identificação de processos e no ajuste de controladores. Será apresentada a base teórica que envolve estes dois procedimentos, assim como, técnicas de adaptabilidade em regime de operação normal. Será apresentada também, uma metodologia de desenvolvimento passo a passo para solução dos problemas em controle, que envolve desde a fase de projeto até a fase de operação e manutenção do sistema.

O quarto capítulo é dedicado ao projeto e simulação do sistema propriamente dito, isto é, o controle de velocidade do motor de corrente contínua utilizando o sistema de controle neuronal. Para isto, é preparado um modelo do acionamento para

simulações em computador. São realizadas as fases de identificação do processo e aprendizado do controlador, sob diversas condições de treinamento para a escolha do método mais eficaz. O sistema construído será testado para várias condições normais e anormais de operação do acionamento. Pretende-se, por exemplo, analisar o comportamento da malha de controle diante da variação de parâmetros e na presença de ruído.

Finalmente, serão apresentadas as conclusões sobre o método empregado, os resultados das simulações e sobre o desempenho do sistema quando comparado com outros tipos de controle.

Tentaremos através deste estudo, verificar se as vantagens que as redes neuronais tem proporcionado em outras áreas também são assimiladas aqui, melhorando ainda mais os sistemas de controle disponíveis. Pretende-se, com isto, dar mais um passo na criação de sistemas de controle auto-ajustáveis com alto nível de confiabilidade.

CAPÍTULO 1

SISTEMAS DE CONTROLE

1.1 - METODOLOGIAS DE CONTROLE

1.1.0 - Introdução

Os sistemas de controle tem tomado conta de muitas das atividades diárias do homem e não apenas na indústria, de tal forma que as pesquisas na área de controle e automação de sistemas tem atraído muito os esforços de pesquisadores independentes e empresas do ramo.

A confiabilidade de um sistema está hoje intimamente associada ao esquema de controle implementado, visto que estes tem expandido suas áreas de atuação a outras antes não exploradas, tais como o controle de qualidade de produtos manufaturados, linhas de montagem automáticas, tecnologia espacial, monitoração e recuperação de sistemas com o auxílio do computador, robótica, sistemas automáticos de transportes e muitas outras que estão surgindo.

1.1.1 - Tipos de sistemas de controle

A forma mais simples de ver um sistema de controle é como mostra a figura 1.1. O objetivo básico do sistema de controle é controlar as saídas de alguma forma predeterminada através das entradas e dos próprios componentes do sistema.

Há algumas estruturas clássicas de sistemas de controle muito conhecidas, como sistema de malha aberta e o sistema com realimentação.

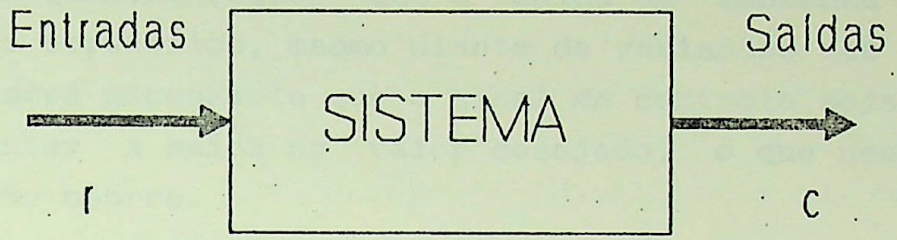


Figura 1.1 - Diagrama básico de um sistema de controle.

O diagrama de blocos da figura 1.2 representa um sistema de controle em malha aberta, que constitui o sistema de controle mais simples. Os elementos podem ser divididos em duas partes : o controlador e o processo controlado. Através do sinal de referência, o bloco controlador encarrega-se de gerar o sinal de controle necessário para o acionamento do processo. O controlador pode ser desde um simples circuito eletrônico ou um pequeno acoplamento mecânico até um sofisticado microcomputador, dependendo da aplicação e das exigências do controle.

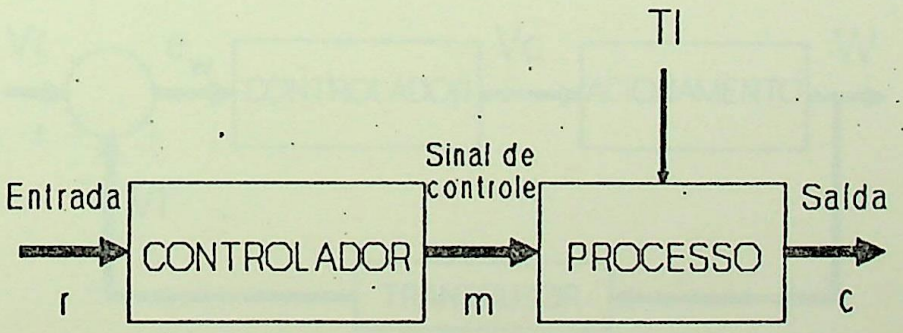


Figura 1.2 - Sistema de controle de malha aberta.

O controle em malha aberta não apresenta um comportamento

satisfatório nos casos em que se exige um mínimo de precisão na variável de saída, o que inclui a grande maioria dos casos. Por exemplo, no diagrama da figura 1.2, um aumento na variável externa *Tl* ocasionará mudanças na variável de saída. Se o sistema em questão requer que a saída se mantenha fixa em um certo valor específico, mesmo diante de variações nas variáveis externas, será necessário que o sinal de controle seja modificado afim de manter a saída no valor desejado, o que neste tipo de controle não ocorre.

Para corrigir esta deficiência é utilizado o sistema com realimentação. O sinal de saída a ser controlado é realimentado e comparado com um valor de referência. Desta comparação é gerado um erro que é utilizado então para corrigir o sinal de controle. O sistema com uma ou mais realimentações é conhecido como sistema de malha fechada. A figura 1.3 mostra um sistema de controle de velocidade em malha fechada. Qualquer variação inesperada na velocidade w é sentida pela realimentação, o que aumentará o valor do erro e_w. O controlador corrige o sinal de controle Vc de acordo com o valor do erro e_w, o que deverá ajustar novamente o valor da velocidade no valor desejado definido pela referência Vr.

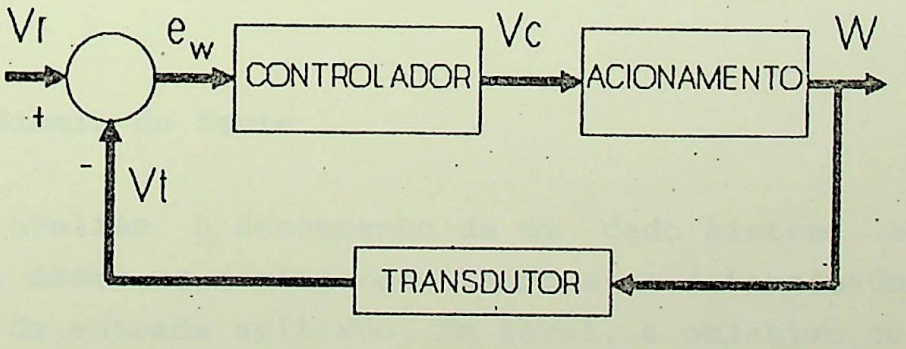


Figura 1,3 - Sistema de controle com realimentação.

O sistema de malha fechada oferece, além de maior precisão

na saída, melhor resposta dinâmica e redução dos efeitos devido a perturbações indesejadas no sistema. Sabe-se também que a realimentação possue grande influência sobre a estabilidade, a sensibilidade e a reação do sistema a estas perturbações externas ou ao ruído.

A implementação do sistema de controle pode ser tanto analógica como digital, sendo que, os sistemas digitais trouxeram inúmeras vantagens às estruturas de controle, o que tornou o uso do computador bastante popular nos últimos anos. Os sistemas de controle digital diferem dos sistemas contínuos pelo fato de que os sinais em um ou mais pontos do sistema estão na forma de um trem de pulsos ou de um código digital. Em geral, um sistema digital recebe dados ou informações de forma intermitente em instantes específicos de tempo.

Deve-se ressaltar que o desempenho do sistema dependerá em grande parte do controlador utilizado, ou melhor, da filosofia de controle empregada.

Neste capítulo mostraremos duas filosofias de controle muito utilizadas devido à sua simplicidade e aplicabilidade.

1.1.2 - Desempenho do sistema

1.1.2.1 - Sinais de teste

Para avaliar o desempenho de um dado sistema estuda-se a resposta do mesmo no domínio do tempo a um determinado sinal de referência de entrada aplicado. Em geral, o objetivo do sistema é fazer com que a variável de saída acompanhe o sinal de entrada, logo é necessário comparar a entrada e a saída como funções do tempo.

O sistema deve ser avaliado de acordo com excitações de entrada reais, ou seja, aquelas que o sistema deverá encontrar

durante o regime de funcionamento normal. Caso tais sinais não sejam conhecidos *a priori* ou tenham um comportamento aleatório, seria necessário definir alguns tipos de entradas básicas para as quais o sistema seria avaliado. Alguns critérios de desempenho podem ser especificados com relação a estes sinais de teste de forma que o sistema seja projetado afim de satisfazer tais critérios. Podemos adiantar que o processo que será objeto de nosso estudo funciona com entradas em degrau e rampas, logo estes poderão ser nossos sinais de teste para avaliação.

1.1.2.2 - Resposta transitória

O desempenho de um sistema pode ser analisado através da sua resposta transitória, a qual é observada para uma entrada em degrau unitário.

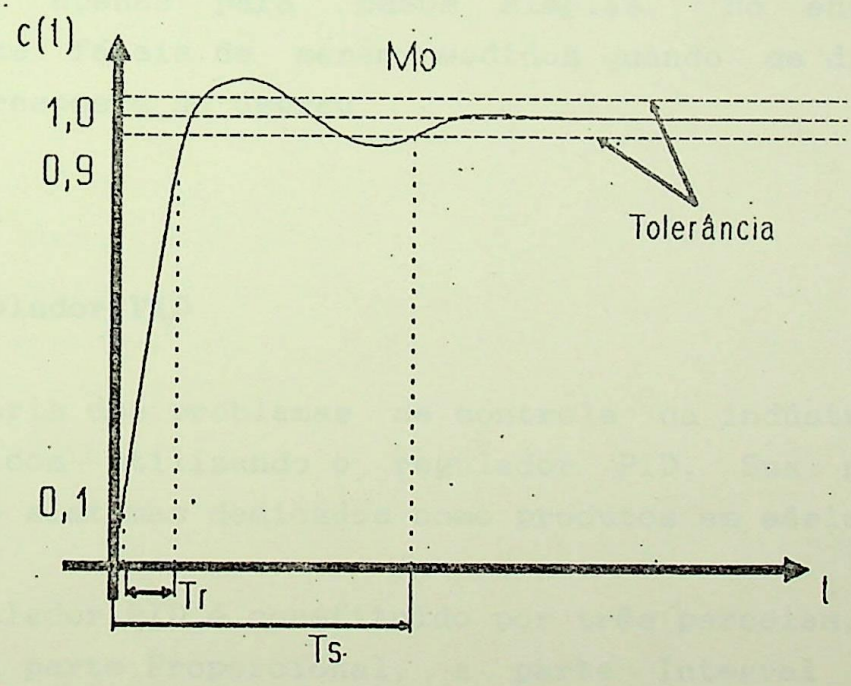


Figura 1.4 - Resposta a uma excitação em degrau unitário.

Alguns critérios definidos em relação à resposta ao degrau, mostrados na figura 1.4, serão do nosso interesse, são eles :

- Sobressinal máximo (*overshoot*) *Mo* : é definido como o maior desvio da saída durante o transitório. É usado também como uma medida da estabilidade relativa do sistema. O sobressinal máximo é representado freqüentemente como uma porcentagem do valor final da resposta ao degrau;

- Tempo de subida *Tr* : é definido como o tempo necessário para a resposta subir de 10 por cento a 90 por cento de seu valor final;

- Tempo de acomodação *Ts* : é definido como o tempo necessário para a resposta ao degrau decrescer e permanecer dentro de uma porcentagem específica de seu valor final. Um valor usado freqüentemente é 5 por cento.

Os três parâmetros definidos dão uma medida direta das características da resposta transitória ao degrau. Estes parâmetros poderiam ser determinados analiticamente, porém, isto é possível apenas para casos simples. No entanto, são relativamente fáceis de serem medidos quando se dispõe de um gráfico da resposta ao degrau.

1.1.3 - Regulador PID

A maioria dos problemas de controle na indústria hoje tem sido resolvidos utilizando o regulador PID. Sua popularidade inclui tanto sistemas dedicados como produtos em série.

O regulador PID é constituído por três parcelas, como indica o nome, a parte Proporcional, a parte Integral e a parte Derivativa, sendo que esta ultima muitas vezes não é necessária. O funcionamento do regulador PID é representado através da equação :

$$f(t) = K \left[e(t) + \frac{1}{Ti} \int e\,dt + Td \frac{de}{dt} \right]$$

ou, em termos de ganhos:

$$f(t) = K_p.e(t) + K_i\int edt + K_d \frac{de}{dt}$$

Aplicando a transformada de Laplace, vem:

$$Gc(s) = K_p + \frac{K_i}{s} + K_d.s \qquad (1.1)$$

O problema do regulador PID consiste no ajuste dos ganhos das três parcelas, o que normalmente é feito por engenheiros com certa experiência utilizando regras simples e empíricas. Tais regras consistem, em geral, na análise da resposta do sistema a degraus na entrada ou a perturbações. Os ganhos do controlador são ajustados até obter um padrão desejado na resposta. Tais ajustes são muitas vezes definidos por parâmetros específicos como o tempo de acomodação, o sobressinal máximo, etc. Mas, estes requerem o conhecimento de experiências prévias ou o resultado de simulações, e isto quando o processo permite.

Uma das técnicas de ajuste mais conhecida e utilizada é a regra de ajuste manual de Ziegler-Nichols, a qual trabalha a partir da chamada freqüência limite de estabilidade. Pelo método de Ziegler-Nichols, o regulador é ajustado para funcionar com apenas a ação proporcional, cujo ganho é aumentado até tornar a saída do sistema de malha fechada instável. O período de oscilação da instabilidade registrada é então utilizado junto com o valor do ganho proporcional correspondente para o cálculo dos três ganhos Kp-Ki-Kd. Os valores dos ganhos são obtidos através das relações mostradas na Tabela 2.1.

Esta experiência é algumas vezes fácil de ser realizada, no entanto, não é recomendada para processos lentos ou com um grande número de perturbações, para os quais torna-se difícil e demorado ou simplesmente levará a resultados insatisfatórios.

TABELA 2.1 - Regras de ajuste Ziegler-Nichols

Tipo	K	Ti	Td
P	0,5 Kc	-	-
PI	0,45 Kc	0,83 Tc	-
PID	0,6 Kc	0,5 Tc	0,125 Tc

As referências [1] e [2] descrevem o projeto, construção e teste de uma malha de controle de velocidade de motor de corrente contínua utilizando o regulador PI, isto é, sem a parte derivativa. [3] estende o mesmo estudo para um motor alimentado por um conversor dual.

1.1.4 - Controle Adaptativo

O surgimento, na indústria, de processos cada vez mais complexos foi sempre o impulso para o desenvolvimento de novas técnicas e metodologias baseadas nas teorias modernas do controle. No entanto, mesmo com os novos avanços, ainda é considerado como requisito de um projeto de controle adequado um bom conhecimento prévio das características do processo a ser controlado.

Os avanços na eletrônica, mais especificamente na área da microeletrônica, dão origem a microprocessadores cada vez menores, mais rápidos e mais poderosos. Além disso, aumentam constantemente os da supervisão, confiabilidade e resposta dos sistemas de controle. A velocidade dos microprocessadores viabiliza a implementação de novas técnicas modernas visando o acompanhamento de exigências cada vez mais rigorosas.

Para eliminar as deficiências do ajuste fixo dos reguladores tipo PID são utilizados os chamados Sistemas de Controle Adaptativos, que visam o controle de processos cujas

características são pouco conhecidas ou desconhecidas, ou quando o processo está sujeito a mudanças internas de parâmetros.

Especificamente, no controle de velocidade de motores de corrente contínua, os parâmetros do motor ou da carga acionada podem variar consideravelmente durante a operação. No caso do regulador PID tais variações poderão causar mudanças no comportamento da resposta dinâmica ou até mesmo a instabilidade da malha. Em [4], os efeitos da variação de parâmetros são analisados para uma malha de controle de posição, onde observa-se na resposta oscilações consideráveis e um overshoot acima do esperado. Obviamente, tais situações não podem ser permitidas em um sistema de controle de posição.

O controle adaptativo foi idealizado precisamente para tentar manter o desempenho do sistema dentro de um padrão especificado sob qualquer condição de funcionamento, mesmo quando da ocorrência de variações esperadas ou não. Em outras palavras, enquanto a realimentação visa a eliminação do efeito das perturbações na saída, o controle adaptativo é orientado à eliminação do efeito das perturbações estruturais no comportamento dinâmico da malha de controle. Tais perturbações estruturais podem ser provocadas por variações tanto de parâmetros do próprio processo como do meio onde ele se encontra.

1.1.4.1 - Estrutura básica

O diagrama de blocos elementar de um sistema de controle adaptativo é mostrado na figura 1.5. O sistema pode ser visto como tendo duas malhas, a malha de realimentação usual, e uma outra malha encarregada do ajuste do controlador.

A malha de ajuste é composta, em geral, por dois blocos, o bloco de Identificação e o bloco de Adaptação. O primeiro cria um modelo através da identificação dos parâmetros do processo. O segundo bloco ajusta os parâmetros do controlador através de certas regras pré-definidas de acordo com os parâmetros do

modelo identificado. As regras de ajuste são definidas visando manter um determinado comportamento da saída do sistema mesmo diante de variações no processo, as quais deverão ser sentidas pelo modelo.

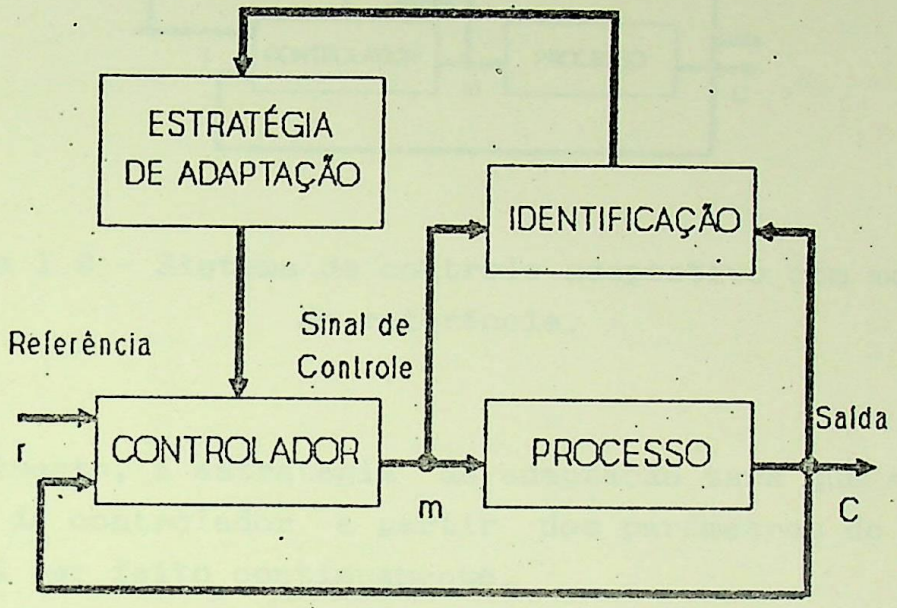


Figura 1.5 - Sistema de controle adaptativo.

Este sistema pode ainda ser otimizado utilizando uma terceira malha, mostrada na figura 1.6. O terceiro bloco a ser incluído, chamado modelo de referência, tem como função especificar as características dinâmicas desejadas na saída do sistema. O sistema da figura 1.6 é conhecido como Sistema de Controle Adaptativo com Modelo de Referência (MRAC - Model Reference Adaptive Control).

De forma geral, o bloco de identificação recebe através de sensores tanto a entrada como a saída do processo afim de ajustar um modelo que deverá, no final do procedimento, representar o melhor possível o processo em estudo. Deverá também, ser capaz de sentir as mudanças que possam ocorrer no processo afim de que modelo as acompanhe.

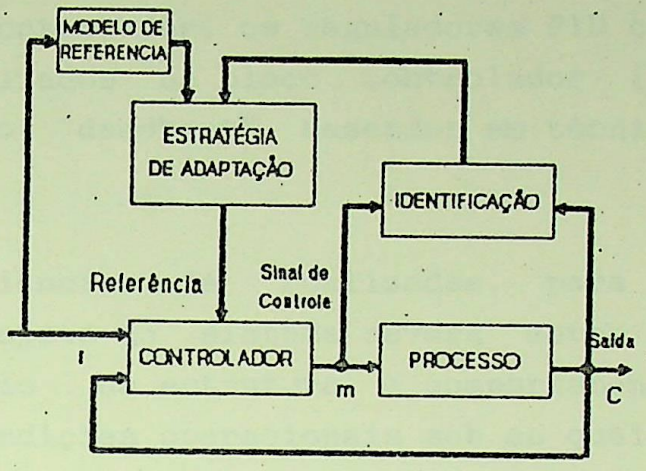


Figura 1.6 - Sistema de controle adaptativo com modelo de referência.

Igualmente, a estratégia de adaptação terá que ajustar os parâmetros de controlador a partir dos parâmetros do modelo, o qual deverá ser feito continuamente.

As configurações das figuras 1.5 e 1.6 são chamadas de Sistemas de Controle Adaptativo Indireto, visto que o controlador é ajustado indiretamente através de regras que dependem de um modelo identificado previamente.

Em alguns casos, o controlador pode ser ajustado diretamente a partir dos próprios parâmetros do modelo obtido. Com isto, o bloco de identificação não é utilizado.

O projeto de um sistema de controle adaptativo consiste na definição do modelo e do algoritmo de identificação a ser implementado, da estrutura do controlador e das regras de adaptação que deverão fazer o elo entre o modelo adotado e o controlador do sistema. São estes detalhes que diferenciam um sistema de outro.

Para o procedimento de identificação o algoritmo mais utilizado é o dos Mínimos Quadrados Recursivo RLS (Recursive Least Squares) devido á facilidade de implementação e velocidade.

Quanto ao controlador; os reguladores PID convencionais tem sido muito utilizados no bloco controlador [5], assim como controladores tipo "deadbeat" baseados em técnicas de resposta rápida [6,7].

Das experiências já realizadas, para garantir bons resultados, o projeto do sistema deverá estar associado a um conhecimento prévio da estrutura e comportamento do processo assim como das condições operacionais sob as quais este opera.

1.1.4.2 - Desvantagens

Até hoje, muitos algoritmos de adaptação tem sido desenvolvidos, vários com sucesso; no entanto, a maioria são complicados demais e poucas aplicações tem sido desenvolvidas com eles, sendo que estas exigem computadores de pequeno e médio porte. Outro detalhe importante é que em muitos dos casos o algoritmo serve para um determinado tipo de aplicação, sendo que para outras aplicações o desempenho cai bruscamente.

No caso do acionamento de motores o uso de computadores pequenos ou médios ainda não é considerado viável, no lugar devem ser usados microprocessadores. Entretanto, a complexidade dos algoritmos ainda constitui um obstáculo para a implementação [8]. Um sistema de controle adaptativo utilizado em acionamento de motores requer rapidez, visto que as rotinas de identificação e ajuste do controlador devem ser feitas "on-line", e isto implica que algoritmos numéricos complexos e sofisticados devam ser executados em alguns poucos milissegundos, dentro do período de amostragem. Requere-se também, para um melhor projeto, informações "a priori" sobre dados como o período de amostragem a ser utilizado, as estruturas dos modelos adotados, e outros.

De qualquer maneira, os sistemas de controle adaptativos representaram um grande passo afim de tornar os sistemas de controle cada vez mais imunes às variações internas e do meio, ao ruído, e, principalmente, cada vez mais independentes do fator

humano.

1.2 - DESCRIÇÃO DO PROCESSO

1.2.0 - Introdução

Existe hoje, um número crescente de processos industriais que requerem exatidão no controle de velocidade, posição e/ou aceleração, assim como uma ampla faixa de variação destas variáveis. Tal é o caso, por exemplo, de injetoras de plástico, laminadores de metais, máquinas operatrizes programáveis e muitos outros sistemas transportadores.

A variação da velocidade, a facilidade do controle e a grande flexibilidade da característica torque-velocidade tem tornado o motor de corrente contínua (CC) o mais utilizado em acionamentos que requerem velocidade variável.

Embora os motores de corrente alternada (CA) ofereçam algumas vantagens, tais como, operar diretamente na rede de alimentação, baixo custo e maior eficácia com menor manutenção, seu uso é indicado normalmente em acionamentos de velocidade constante.

O motor de corrente contínua mantém um alto rendimento em toda a sua faixa de atuação da velocidade e oferece maiores facilidades para a inversão do sentido de rotação, mesmo em motores de grande porte com cargas de inércia alta [9].

O controle de velocidade pode ser realizado facilmente através da tensão de armadura da máquina, mantendo-se a excitação constante. Isto é válido para velocidades entre zero e a velocidade nominal. Acima da nominal, o controle é feito alterando-se o fluxo de excitação.

Entre os tipos de motor CC,o mais utilizado é o motor com

excitação independente, visto que os circuitos da armadura e do campo são alimentados por fontes independentes, ele permite uma maior flexibilidade no controle.

Antigamente, para acionar o motor era utilizado um conjunto motor-gerador, conhecido como Ward-Leonard. Este conjunto possui um motor auxiliar girando a velocidade constante acionando um gerador. Variando a excitação do campo do gerador, varia-se a tensão na saída deste, que por sua vez alimenta o motor principal, permitindo o controle contínuo de sua velocidade.

Porém, este sistema, já ultrapassado, além de ter um custo muito elevado, resposta lenta e precisão baixa, requer muito espaço e manutenções periódicas, apresentado contudo um rendimento no máximo de 80%.

Os grandes avanços conseguidos no campo dos semicondutores de potência, dos circuitos integrados e, ultimamente, dos microprocessadores, que tem tornado tais produtos industrializáveis em grande escala, permitiram a adoção de sistemas eletrônicos direcionados ao acionamento de máquinas. Desta forma, hoje em dia, o grupo Ward-Leonard é substituído por conversores estáticos, ou seja, aqueles que utilizam semicondutores de potência.

Os acionamentos estáticos oferecem as seguintes vantagens:

- alta eficiência na potência transmitida;
- controle através de sinal de baixa potência;
- requer pouca manutenção;
- pequeno volume e pouco peso, trazendo custos de instalação e operação menores.

Entre os semicondutores de potência, mais especificamente os tiristores, o mais flexível é o SCR (Silicon Controlled Rectifier), que consegue dissipar grandes quantidades de calor, sendo portanto, utilizado no controle de máquinas e na transmissão de corrente contínua, além de inúmeras outras aplicações que envolvem elevadas correntes e/ou elevadas tensões.

Deve-se ressaltar, no entanto, que o uso de um acionamento tiristorizado traz consigo duas desvantagens que devem ser consideradas :

- a saída do conversor possui um "ripple", suficientemente alto pra provocar aquecimento e problemas de comutação. Para aliviar tais efeitos é colocado um indutor de alisamento no circuito da armadura;

- sob certas condições de operação, o fator de potência da alimentação AC é baixo.

O acionamento de um motor de corrente contínua através do conversor tiristorizado pode ser esquematizado pelo diagrama representativo da figura 1.7.

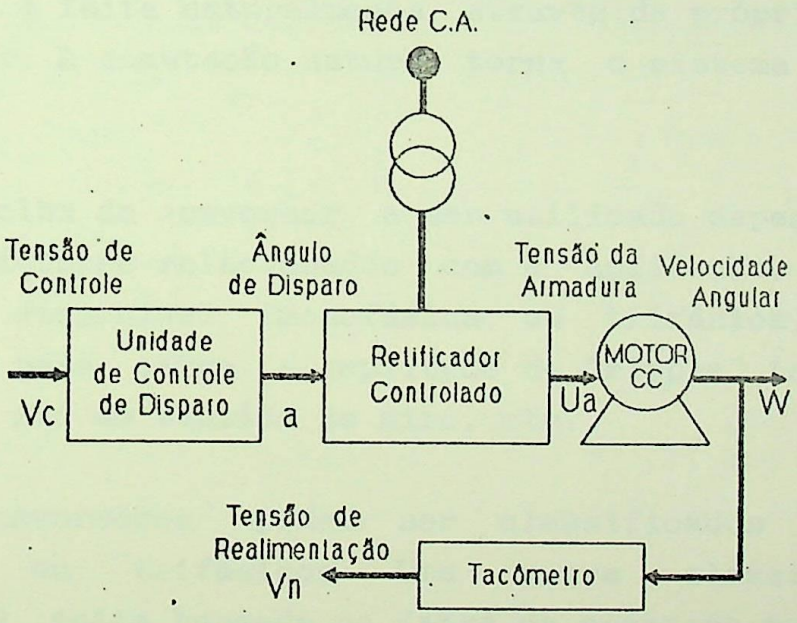


Fig.1.7 - Diagrama de blocos representativo do processo.

De acordo com a figura 1.7, o acionamento pode ser dividido em três estágios, que são :

- circuito de disparo
- conversor tiristorizado e,
- motor CC

No eixo do motor é acoplado um tacogerador para efeito de realimentação da velocidade, constituindo parte, portanto, do terceiro estágio. Cada estágio será analisado nos próximos itens deste capitulo, sendo que, para seguir uma seqüência lógica, estudaremos primeiro o conversor.

1.2.1 - Conversor tiristorizado

O conversor é constituído basicamente por diodos e/ou tiristores, e tem como objetivo básico converter a tensão alternada de entrada em uma tensão de saída contínua variável.

Os conversores, que serão mostrados aqui, são os do tipo com comutação pela rede. Nestes conversores, a comutação entre os tiristores, ou melhor, a transferência de corrente entre os tiristores, é feita naturalmente através da própria alimentação do conversor. A comutação natural torna o sistema simples e mais barato.

A escolha do conversor a ser utilizado depende diretamente de vários fatores relacionados com a aplicação, entre eles, a alimentação disponível (monofásica ou trifásica), a potência requisitada pela carga, a amplitude de "ripple" tolerável, se há reversão ou não do sentido de giro, etc..

Os conversores podem ser classificados basicamente em monofásicos ou trifásicos. Uma outra classificação mais específica é feita baseada na faixa de operação no plano Ua-Ia, isto é, tensão vs corrente de armadura do motor, as quais coincidem com a tensão v_{da} e a corrente I_d fornecidas pelo conversor. Assim, quando a carga acionada é um motor CC, o conversor pode ser classificado de acordo com o quadrante de funcionamento segundo a figura 1.8.

Conversores semi-controlados, por exemplo, operam no quadrante I, isto significa que a corrente e a tensão operam com uma polaridade só. Em termos de potência, pode-se

dizer que o fluxo ocorre sempre do lado CA para o lado CC, da rede de alimentação para o motor.

Já os conversores controlados podem operar nos quadrantes I e IV, onde a polaridade da tensão da armadura pode ser invertida. No entanto, a corrente circula sempre no mesmo sentido, já que o SCR, como o diodo, conduz em um único sentido.

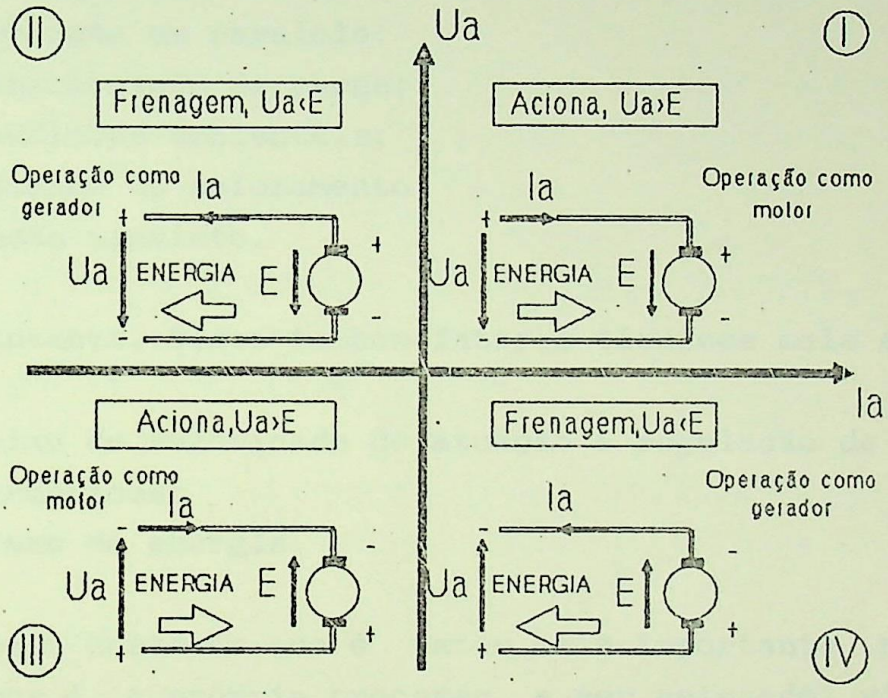


Figura 1.8 - Classificação do conversor de acordo com a faixa de operação do acionamento.

Quando é necessária a reversão de velocidade é utilizado o chamado conversor dual que opera nos quatro quadrantes. O conversor controlado consiste de dois conversores controlados ligados em anti-paralelo alimentados por um único transformador. Consegue-se através dele controlar a velocidade nos dois sentidos de giro, além de permitir a regeneração de energia, isto é, a devolução de energia da máquina para a rede CA.

1.2.1.1 - Escolha do conversor

A correta escolha do conversor envolve, como já foi mencionado no item anterior, a análise de fatores técnicos que, quando estudados cuidadosamente, permitem um melhor aproveitamento do equipamento a um custo mais baixo.

Segundo [10], no procedimento de seleção deve-se levar em conta, entre outros fatores :

- as características do processo a ser acionado;
- operação em paralelo;
- distribuição da carga;
- condições ambientais;
- tamanho do acionamento;
- custo previsto.

No entanto, temos também fatores técnicos mais específicos, tais como :

- faixa de velocidade de atuação e regulação de velocidade;
- harmônicos;
- fluxo de energia.

Deve-se destacar que o fator mais importante na definição do conversor é o próprio processo a ser acionado, visto que, o desempenho do sistema como um todo dependerá muito da qualidade do acoplamento acionador-carga. Entretanto, o estudo sério e cuidadoso dos outros fatores facilita a implementação e aplicação de outras atividades inerentes ao processo, como: a manutenção e proteção dos equipamentos, o controle de qualidade e as futuras modificações e/ou ampliações do acionamento.

Neste ponto devemos lembrar as vantagens e desvantagens dos sistemas que utilizam motores de corrente contínua. Um acionamento disponível mais simples e confiável é mais um aspecto que destaca o motor CC em aplicações de velocidade variável.

Para fins deste desenvolvimento, como o nosso objetivo principal não é o projeto de um acionamento, mais sim a análise do controle, utilizaremos um acionamento simples, sem necessidade

de reversão da velocidade, e de fácil manuseio matemático. Podemos optar então entre o conversor semi-controlado e o conversor controlado. No entanto, o conversor controlado possue menos requisitos quanto a filtragem da saída e oferece menos problemas devido a distorções no lado C.A. devido à presença de harmônicos de menor ordem.

1.2.1.2 - Ponte trifásica tiristorizada

O circuito conhecido como ponte trifásica tiristorizada, mostrado na figura 1.9, é o conversor mais utilizado no controle de motores CC.

O circuito em questão é um conversor do tipo com comutação pela rede, onde a comutação dos tiristores é processada via transformador de alimentação.

Na verdade, o conversor inclui tanto a ponte, isto é, o circuito formado pelos SCR's, como o circuito de disparo.

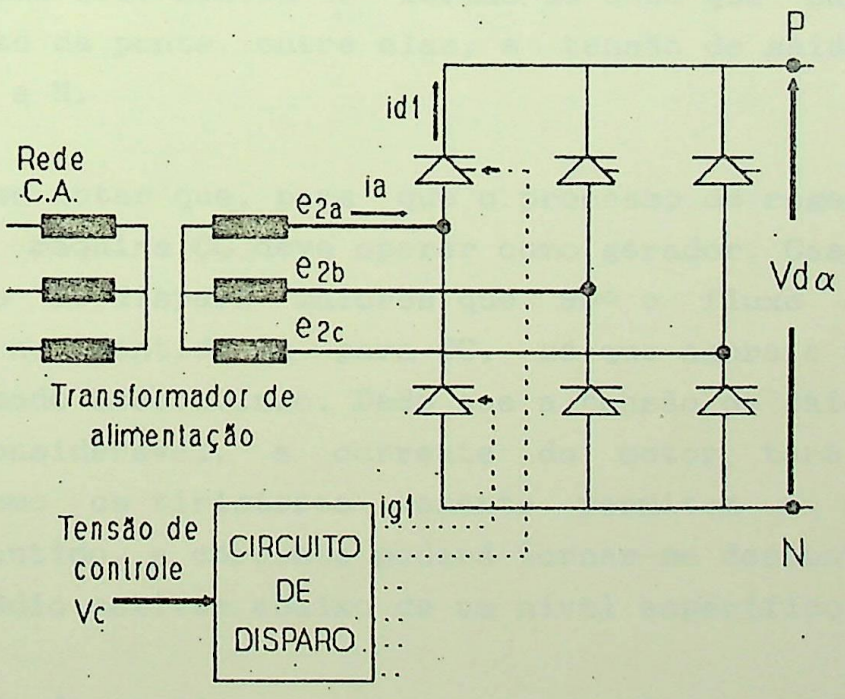


Figura 1.9 - Circuito da ponte trifásica tiristorizada.

O circuito de disparo será analisado no item 1.2.2, por enquanto nos ocuparemos apenas do circuito de potência do conversor.

A ponte pode operar de duas maneiras, como retificadora e como inversora. Operando como retificadora, a ponte faz a conversão de potência CA da rede de alimentação para potência CC que será entregue ao motor. Neste tipo de operação, o ângulo de disparo α varia entre 0° e 90°.

O modo de operação como inversora ocorre para ângulos de disparo entre 90° e 180°. Neste modo, a tensão de saída da ponte poderá tornar-se negativa, ocorrendo o processo de regeneração. Mas, para que isto seja possível, a tensão de armadura do motor deve ser invertida através de contatores ou pode-se também inverter o sentido da corrente de excitação do motor.

Da figura 1.9, vemos que a tensão de saída v_{da} é obtida entre polos P e N de retificação. A tensão v_{da} varia, de acordo com o ângulo de disparo, entre zero e um valor máximo que será definido pelo equacionamento da tensão de saída da ponte.

A figura 1.10 mostra as formas de onda que caracterizam o funcionamento da ponte, entre elas, a tensão de saída gerada nos terminais P e N.

Deve-se notar que, para que o processo de regeneração seja possível, a máquina CC deve operar como gerador. Caso contrário, para ângulos de disparo maiores que 90° o fluxo de potência permanecerá no sentido CA para CC, só que agora a ponte poderá operar no modo descontínuo. Dado que a tensão de saída possui um "ripple" considerável, a corrente do motor terá também um "ripple". Como os tiristores somente permitem a condução em apenas um sentido, a corrente poderá tornar-se descontínua quando seu valor médio estiver abaixo de um nível específico.

O modo descontínuo ocorre mais freqüentemente com cargas pequenas e velocidade de giro elevadas.

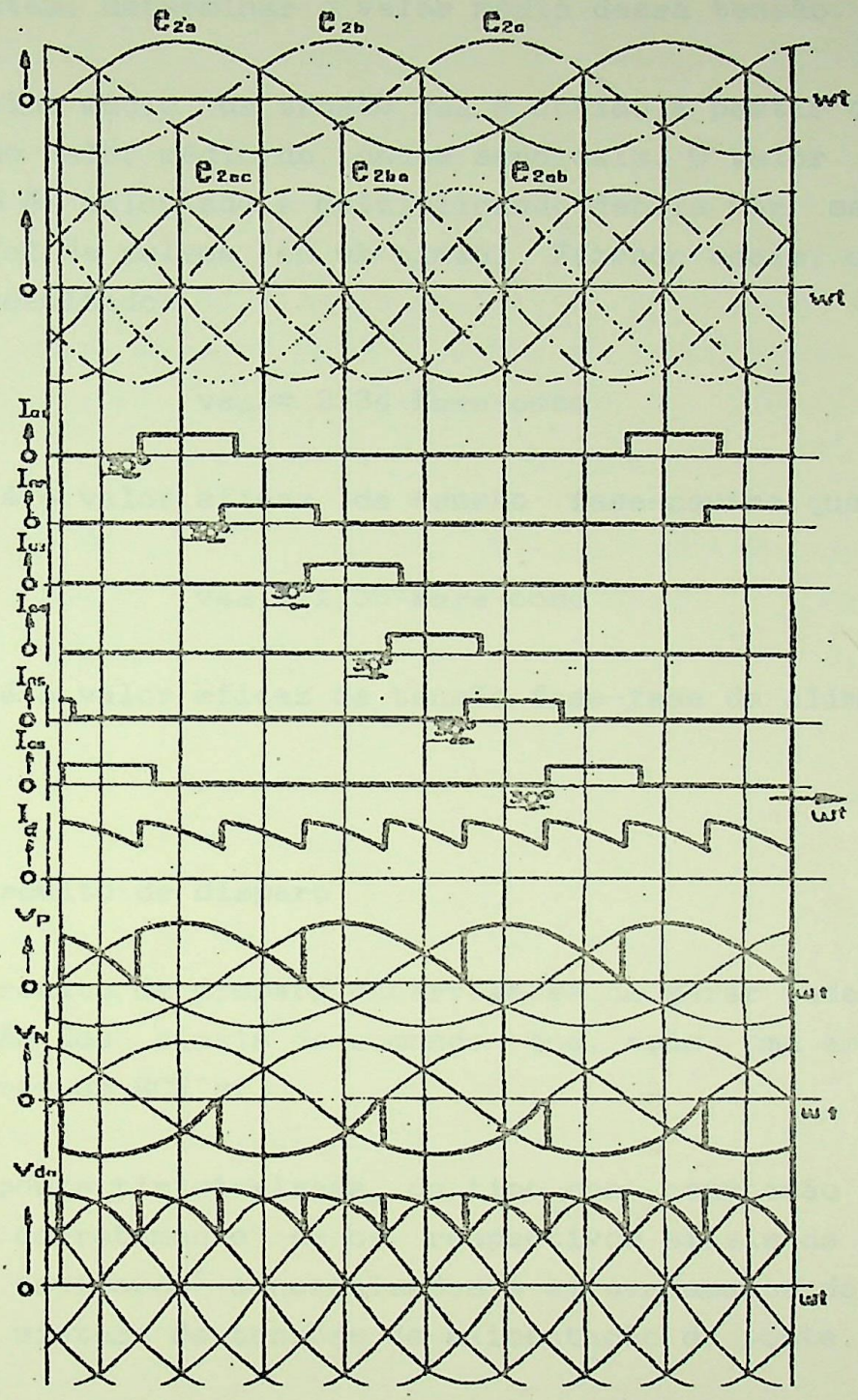


Figura 1.10 - Funcionamento da ponte trifásica tiristorizada.

1.2.1.3 - Tensão de saída do conversor

Assumindo que a corrente do motor é contínua, a tensão v_{da} fornecida pela ponte depende apenas do ângulo de disparo α.

Podemos então, determinar o valor médio dessa tensão.

O valor médio da tensão $v_{d\alpha}$ é obtido a partir da teoria de cálculo do valor médio de ondas senoidais. O valor médio de um dos pulsos é calculado e multiplicando depois por seis, que é o número total de pulsos em um ciclo. Fazendo assim, chega-se ao seguinte resultado:

$$v_{d\alpha} = 2{,}34 \cdot E_{2fn} \cdot \cos\alpha \qquad (1.2)$$

onde E_{2fn} é o valor eficaz da tensão fase-neutro que alimenta a ponte, ou

$$v_{d\alpha} = 1{,}35 \cdot E_{2ff} \cdot \cos\alpha \qquad (1.3)$$

onde E_{2ff} é o valor eficaz da tensão fase-fase da alimentação.

1.2.2 - Circuito de disparo

O circuito de disparo encarrega-se de gerar toda a lógica de formação dos sinais de comando que, após uma amplificação, irão disparar os SCR's.

Uma ponte tiristorizada, do tipo com comutação pela rede, funcionará corretamente se os respectivos sinais de disparo dos tiristores estiverem sincronizados e adequadamente defasados com relação ao sistema de tensões de alimentação da ponte.

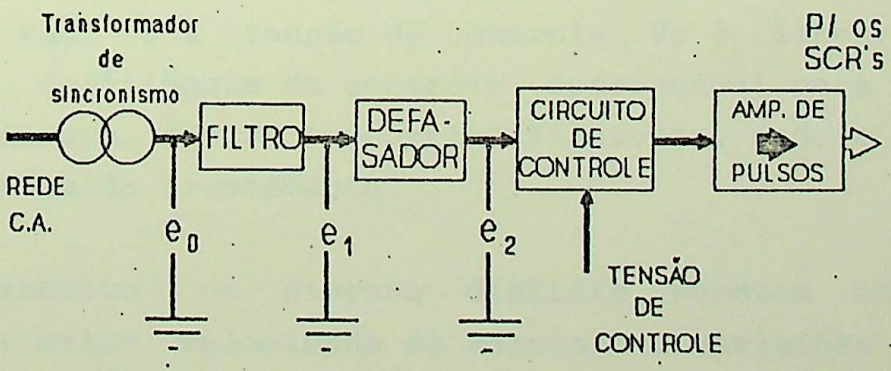


Figura 1.11 - Diagrama de blocos do circuito de disparo.

A figura 1.11.mostra o diagrama de blocos básico de um circuito de disparo. O filtro, do tipo passa-baixa, é utilizado para deixar passar apenas a componente fundamental, perfeitamente senoidal, para o circuito defasador. O defasador tem como finalidade adiantar ou atrasar a tensão de saída do filtro, de tal modo a existir uma sincronização correta com a alimentação da ponte. A defasagem depende da filosofia de disparo adotada. Em outras palavras, os dois primeiros blocos apresentados constituem circuitos destinados à preparação do sinal de sincronismo.

Os sinais de disparo são gerados através do sinal de sincronismo e da tensão de controle Vc, através de uma lógica que depende do tipo de circuito implementado, representado no diagrama pelo pelo terceiro bloco da figura 1.11.

1.2.2.1 - Filosofia de disparo

São conhecidas várias filosofias de circuitos de disparo, as quais são divididas, básicamente, em circuitos analógicos e circuitos digitais.

Dentro dos circuitos analógicos temos duas filosofias muito utilizadas, o controle linear ou também chamado de rampa tipo 1, e o controle de tipo cossenoidal.

O controle cosenoidal além de apresentar um circuito mais simples de ser implementado pode também ser analisado mais fácilmente, visto que a relação entre a tensão de saída do conversor, $v_{d\alpha}$, e a tensão de controle Vc é linear. Mas, vale destacar a deficiência do controle cossenoidal para ângulos de disparo pequenos, devido à própria filosofia, onde a precisão do ângulo é reduzida bruscamente.

Os circuito de disparo digitais possuem uma vantagem importante, maior velocidade de resposta a variações no sinal de referência do sistema de controle.

Com o avanço na microeletrônica, tanto na diminuição dos circuitos integrados como no aumento da velocidade de resposta dos mesmos, além do disparo dos tiristores um circuito à base de microprocessador é capaz de realizar outras funções, tais como, diagnóstico e correção de falhas e controle de malha fechada [11].

Uma desvantagem intrínseca aos circuitos de disparo digitais é que o controle de ângulo de disparo é discreto e não contínuo, como nos circuitos analógicos. Com isto, a resolução obtida depende diretamente do número de bits utilizados pelo microprocessador. No entanto, com a tecnologia atual a resolução obtida é muito boa, perto de 0,7° para microprocessadores de 8 bits, podendo ser aumentada ainda mais utilizando microprocessadores com maior número de bits [12].

Para simplificar nosso estudo, pelo mesmo motivo especificado na escolha do conversor, a filosofia de disparo a ser adotada é a do tipo cossenoidal, que será mostrada a seguir.

1.2.2.2 - Circuito tipo cossenoidal

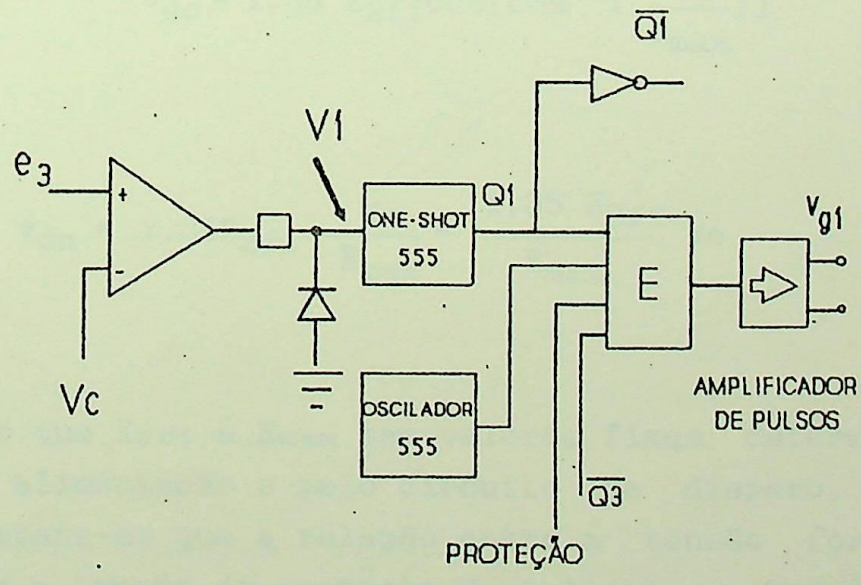


Figura 1.12 - Circuito de disparo tipo cossenoidal.
(Esquema para um SCR)

A figura 1.12 mostra o esquema básico de implementação do circuito de disparo tipo cossenoidal.

A tensão e_2, após o defasador, deve estar adiantada de 60° em relação a tensão e_1 da figura 1.11. O disparo é gerado pela comparação do sinal e_2 com a tensão de controle Vc :

$$E_{max} \cdot \cos\alpha = Vc \qquad (1.4)$$

onde E_{max} ----> valor de pico da onda cossenoidal. É o valor da tensão de controle Vc correspondente a um ângulo de disparo $\alpha = 0°$.

Assim, o ângulo de disparo será dado por:

$$\alpha = \cos^{-1}\left(\frac{Vc}{E_{max}}\right) \qquad (1.5)$$

Lembrando a equação (1.3), do conversor tiristorizado :

$$v_{d\alpha} = 1{,}35 \cdot E_{2ff} \cdot \cos\alpha \ [V]$$

Levando (1.5) e (1.4) em (1.3), teremos:

$$v_{d\alpha} = 1{,}35\ E_{2ff} \cos\left[\cos^{-1}\left(\frac{Vc}{E_{max}}\right)\right]$$

$$v_{d\alpha} = 1{,}35 E_{2ff} \frac{Vc}{E_{max}} = \frac{1{,}35\ E_{2ff}}{E_{max}} Vc \qquad (1.6)$$

Sendo que E_{2ff} e E_{max} tem valores fixos determinados pelo sistema de alimentação e pelo circuito de disparo, respectivamente, constata-se que a relação entre a tensão fornecida pelo conversor e a tensão de controle Vc é linear, e tal linearidade é dada pela constante:

$$Kc = \frac{1,35 \cdot E_{2ff}}{E_{max}} \qquad (1.7)$$

O sinal de saída do circuito de disparo pode não ser suficientemente forte para disparar o SCR. Portanto, é utilizado um amplificador de pulso para servir de acoplamento entre o circuito de disparo e os terminais do SCR. O transformador de pulsos serve também como isolamento entre os dois circuitos.

1.2.3 - Motor de Corrente Contínua

Como indicado no início do capítulo, trataremos apenas com o motor CC com excitação independente. A figura 1.13 mostra o circuito equivalente do motor, a partir do qual serão obtidas as equações básicas que determinam seu funcionamento.

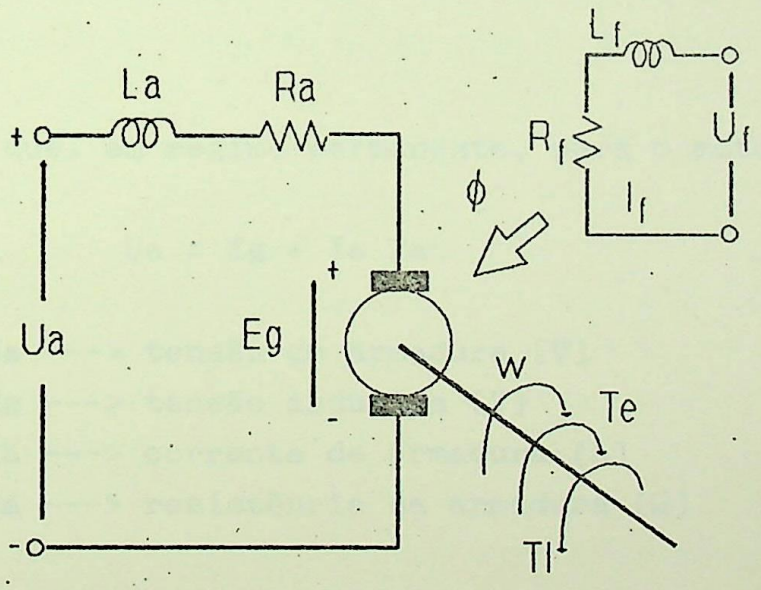


Figura 1.13 - Circuito equivalente do motor CC.

1.2.3.1 - Funcionamento do motor

A figura 1.14 apresenta algumas características do motor CC com excitação independente. A região entre a velocidade nula e a

velocidade nominal é chamada de *"região de torque constante"*, enquanto que, entre a velocidade nominal e a velocidade máxima é chamada de *"região de potência constante"*.

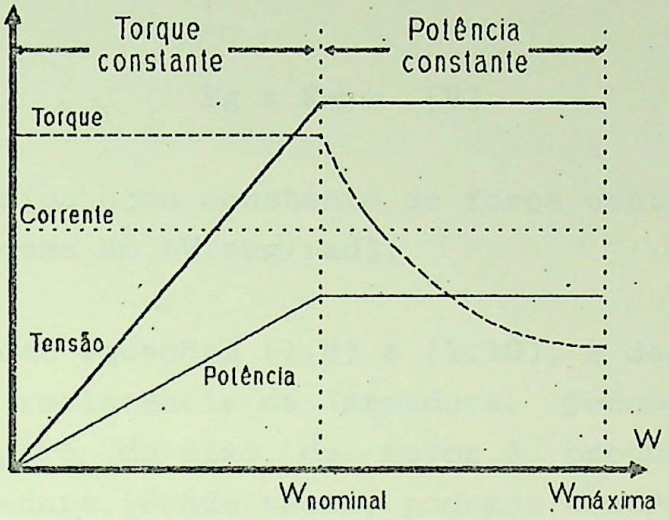


Figura 1.14 - Característica do motor CC.

Sabe-se que, em regime permanente, para o motor CC, tem-se:

$$Ua = Eg + Ia \cdot Ra \qquad (1.8)$$

onde Ua ---> tensão de armadura [V]
Eg ---> tensão induzida [V]
Ia ---> corrente de armadura [A]
Ra ---> resistência da armadura [Ω]

E sabe-se também que a tensão induzida no motor, Eg, também chamada de força contra-eletromotriz (fcem), é proporcional ao fluxo criado pela corrente do campo e à velocidade de rotação w, isto é,

$$Eg \propto \phi \cdot w \longrightarrow Eg = Ke \cdot \phi \cdot w \qquad (1.9)$$

O fluxo de excitação ϕ depende da corrente de excitação i_f,

e a relação entre os dois é não-linear devido ao efeito de saturação presente no circuito magnético do motor. No entanto, dentro da região de torque constante, a corrente do campo é mantida constante, logo então, o fluxo mantém-se também constante. Desta forma, a tensão Eg dependerá apenas da velocidade de giro w:

$$Eg = K\phi \cdot w \quad [V] \qquad (1.10)$$

onde $K\phi$ é conhecida como constante de força contra-eletromotriz, e pode ser expressa em [V.seg/rad].

A partir das equações (1.8) e (1.10), e desprezando a queda de tensão na resistência da armadura, podemos afirmar que a velocidade de giro do eixo do motor é proporcional à tensão aplicada na armadura. Sendo assim, podemos controlar a velocidade do motor a partir da variação da tensão da armadura. Se a corrente da armadura é mantida constante o torque desenvolvido pelo motor será também constante. É por isto que esta região de trabalho é chamada de "*região de torque constante*".

Normalmente, utiliza-se o controle de velocidade através da tensão de armadura para velocidades até o valor nominal. Para velocidades acima da nominal, o controle é realizado através do fluxo de excitação. Isto é, para girar a velocidades acima do valor nominal enfraquece-se o campo. Como a corrente e a tensão da armadura são mantidas constantes, esta região é chamada de "*região de potência constante*".

Dependendo da aplicação, pode ser necessário o controle apenas acima ou abaixo da velocidade nominal, o que simplificaria muito o sistema de controle. Já nos casos em que se exige o controle em toda a faixa de operação da máquina, o sistema de controle deve ter acesso tanto à tensão da armadura quanto à corrente do campo para controle do fluxo.

Neste trabalho trataremos apenas com velocidades de giro até o valor nominal, ou seja, o controle de velocidade será realizado através da tensão de armadura do motor.

1.2.3.2 - Modelagem

A partir do circuito equivalente da figura 1.13 podemos escrever a primeira equação básica do motor :

$$Ua = Eg + Ra \cdot Ia \cdot La \frac{dIa}{dt} \quad [V] \qquad (1.11)$$

onde La ——> indutância (componente indutiva) da armadura [H].

A segunda equação básica é a que relaciona a força contra-eletromotriz com a rotação, dada pela equação (1.10).

Da teoria básica das máquinas de corrente contínua, sabe-se que o torque elétrico desenvolvido pelo motor é proporcional ao fluxo e à corrente de armadura, isto é :

$$Te = Kt \cdot \phi \cdot Ia \quad [N.m] \qquad (1.12)$$

Mas, sendo que, abaixo da velocidade nominal o fluxo é mantido constante, teremos :

$$Te = Kt\phi \, Ia \quad [N.m] \qquad (1.13)$$

onde $Kt\phi$ é chamada de constante de torque, e pode ser expressa em [N.m/A].

A quarta equação característica do motor é a equação de balanço do torque :

$$Te = B \cdot w + J\frac{dw}{dt} + T_l \quad [N.m] \qquad (1.14)$$

onde B ---> coeficiente de fricção do conjunto motor-carga [N.m.seg/rad].

J ---> inércia do conjunto motor-carga [kg.m²].

T_l---> torque da carga [N.m].

Obs.: Pode ser demonstrado que a constante de torque e a constan-

te de força contra-eletromotriz são numericamente iguais, assim, para simplificar a análise faremos :

$$Kt\phi = Ke\phi = Ka\phi$$

lembrando apenas que a igualdade ocorre apenas em magnitude, sendo que as respectivas unidades devem ser respeitadas.

Tomando a transformada de Laplace das quatro equações básicas (1.10), (1.11), (1.13) e (1.14), utilizando as diretivas do Apêndice A, temos :

$$Ua(s) = Eg(s) + Ra \cdot Ia(s) + La \cdot s \cdot Ia(s) \qquad (1.15)$$

$$Eg(s) = Ka\phi W(s) \qquad (1.16)$$

$$Te(s) = T_1(s) + B \cdot W(s) + J \cdot s \cdot W(s) \qquad (1.17)$$

$$Te(s) = Ka\phi Ia(s) \qquad (1.18)$$

Modificando a equação (1.15) :

$$Ia(s) = \frac{Ua(s) - Eg(s)}{Ra + s \cdot La} = \frac{[Ua(s) - Eg(s)] / Ra}{1 + \tau_e \cdot s} \qquad (1.19)$$

onde $\tau_e = La / Ra$ é conhecida como constante de tempo elétrica do motor.

Modificando agora a equação (1.17) :

$$W(s) = \frac{T(s) - T_1(s)}{B + J \cdot s} = \frac{[T(s) - T_1(s)] / B}{1 + \tau_m s} \qquad (1.20)$$

onde $\tau_m = J / B$ é conhecida como constante de tempo mecânica do motor.

As duas últimas equações (1.19) e (1.20), junto com as equações (1.16) e (1.18) podem ser mostradas em forma de diagrama de blocos na figura 1.15. A partir deste diagrama pode ser analisada a variação da velocidade de giro do motor diante de variações na tensão de armadura Ua ou no torque da carga $T_1(s)$. A resposta da velocidade a tais variações é determinada

separadamente e depois combinada de acordo com o princípio da superposição.

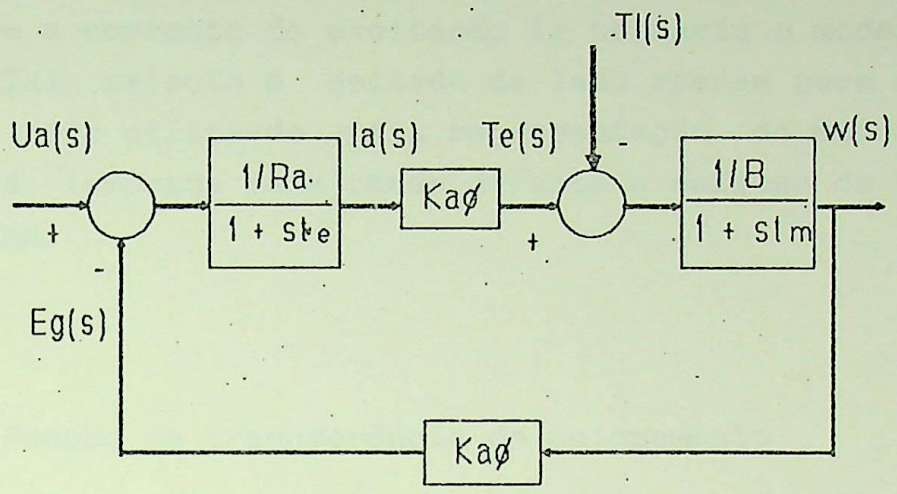


Figura 1.15 - Diagrama de blocos do motor CC.

Neste trabalho nos ocuparemos apenas com a resposta do sistema diante de variações na tensão da armadura. Sendo assim, a parcela que introduz as perturbações no torque da carga é considerada nula, $T_l(s) = 0$. Desta forma, a partir da figura 1.15 podemos obter a seguinte função de transferência :

$$\frac{W}{Ua}(s) = \frac{K_a\phi}{K_a\phi^2 + Ra \cdot B \cdot (1 + s\tau_e)(1 + s\tau_m)} \qquad (1.21)$$

Se $\tau_e << \tau_m$, o que ocorre na maioria dos casos [2], podemos desprezar τ_e, simplificando a função de transferência (1.21) para:

$$\frac{W}{Ua}(s) = \frac{K_a\phi}{K_a\phi^2 + Ra \cdot B + s \cdot Ra \cdot B \cdot \tau_m} \qquad (1.22)$$

Apresentando (1.22) de outra forma :

$$\frac{W}{Ua}(s) = \frac{K_a\phi / (Ra \cdot B)}{s \cdot \tau_m + [1 + K_a\phi^2 / (Ra \cdot B)]} \qquad (1.23)$$

A função de transferência (1.23) será utilizada mais tarde

como modelo para simulação do comportamento dinâmico do motor na malha de controle.

Deve-se ressaltar também que a relação não-linear entre o fluxo ϕ e a corrente de excitação i_f tornaria o modelo não-linear também. Tal relação é deixada de lado apenas para simplificar o modelo a ser utilizado para representação do motor, no entanto esta será lembrada mais tarde durante a análise de comportamento do sistema.

1.2.4 - Função de transferência do acionamento

O processo adotado para o estudo da funcionalidade das redes neuronais na malha de controle foi apresentado, em termos de diagrama de blocos, na figura 1.7. A seguir, cada um dos blocos que o compõem foi apresentado, matematicamente, em termos da função de transferência de cada um dos blocos.

Neste item, agora, pretende-se juntar as equações que definem cada um dos blocos para construir uma função de transferência que represente o funcionamento do processo completo.

No item 1.2.2.2 foi obtida uma relação entre a tensão de controle Vc que alimenta o circuito de disparo e a tensão v_{da} gerada pelo conversor dada pela equação (1.7):

$$\frac{v_{da}(s)}{Vc(s)} = \frac{Ua(s)}{Vc(s)} = Kc$$

E, da mesma forma, no item 1.2.3.2, foi obtida a função de transferência do motor na equação (1.23).

O tacogerador acoplado ao eixo do motor, no papel de transdutor, fornece uma tensão proporcional à velocidade de giro do motor, logo, é representado, em termos de diagrama de blocos através de uma constante ou um ganho, que denominamos Kt. O valor

de Kt é determinado por parâmetros físicos do próprio transdutor.

Juntando os três blocos acima relacionados obtemos o diagrama de blocos completo do processo na figura 1.16, do qual, através da matemática dos blocos podemos obter a seguinte função de transferência :

$$\frac{Vt}{Vc}(s) = \frac{Kc \cdot [Ka\phi/(Ra \cdot B)] \cdot Kt}{s\tau_m + [1 + Ka\phi^2 / (Ra \cdot B)]} \qquad (1.24)$$

A função de transferência completa (1.24) será utilizada mais tarde para simular o comportamento do processo na malha de controle de velocidade.

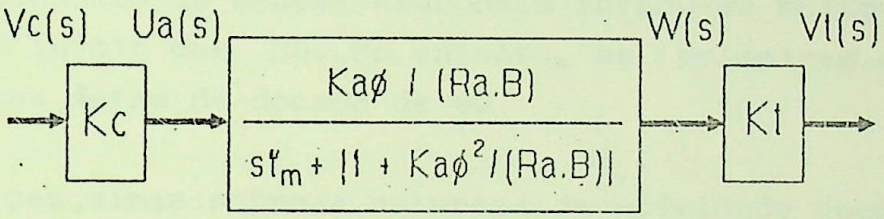


Figura 1.16 - Diagrama de blocos completo do processo.

CAPÍTULO 2

CONCEITOS GERAIS SOBRE REDES NEURONAIS

2.1 - HISTÓRICO

2.1.1 - Antecedentes

O conceito de Redes Neuronais tornou-se muito popular nos anos 80 e início dos 90. No entanto, as primeiras contribuições importantes datam da década de 40.

As pesquisas sobre a natureza da atividade neuronal do ser humano durante os anos 40 resultaram em conceitos e teorias significativas sobre a estrutura e funcionamento biológico das redes neuronais. Partindo destes estudos surgiu o chamado movimento "cibernético", o qual visava desenvolver a analogia entre os conceitos biológicos com os da engenharia e da matemática. A cibernética procurou a criação de esquemas e estruturas capazes de realizar funções até então atribuídas apenas ao cérebro humano, entre elas, a capacidade de reconhecer, acompanhar, memorizar e aprender.

No artigo "A Logical Calculus of the Ideas Immanent in Nervous Activity", Warren McCulloch e Walter Pitts apresentam nesta época um dos primeiros trabalhos sobre computação e modelagem neuronal. Nele McCulloch e Pitts colocam a primeira discussão sofisticada sobre sistemas neuronais baseados no conhecimento biológico do início dos anos 40. Tal discussão tornou-se uma das bases para o desenvolvimento da implementação digital de procedimentos computacionais baseados no modelo

neuronal.

Em 1947, McCulloch e Pitts publicam um segundo ensaio titulado "How We Know Universals", no qual são descritas arquiteturas em rede capazes, em princípio, do reconhecimento de padrões.

A década termina com a publicação do livro de Donald Hebb "The Organization of Behavior", em 1949, considerado como o primeiro a definir uma regra de atualização de parâmetros de aprendizagem, conhecida como regra de Hebb.

Durante a década de 50 e início dos anos 60, aumentou o interesse e cresceram os esforços destinados ao desenvolvimento de máquinas que pudessem "aprender", baseados nos modelos de processamento paralelo. Talvez, uma das primeiras máquinas com esse objetivo foi construída por Marvin L. Minsky em 1951. A máquina de Minsky era baseada no princípio das "*ações reforçadas*", e era constituída de unidades eletrônicas da época, as quais eram interligadas através de uma rede de conexões. A cada uma das conexões era atribuído uma probabilidade ajustável de recepção dos sinais de ativação, sinais que eram transmitidos por sua vez para as outras unidades. Através das "*ações reforçadas*" cada decisão sobre o comportamento da máquina era traduzido em pequenas mudanças das probabilidades, tanto na magnitude como no sinal correspondente [13].

Em 1958, Frank Rosenblatt definiu uma estrutura de rede neuronal chamada *Perceptron*. O *Perceptron*, segundo Rosenblatt, é capaz de aprender a classificar conjuntos de padrões através da modificação das conexões internas.

Em 1960, Bernard Widrow e M. E. Hoff publicaram o artigo "Adaptive Switching Circuits". Widrow e Hoff projetaram ferramentas de redes neuronais que simularam em computador e ainda implementaram em hardware. Foram eles que introduziram o dispositivo conhecido hoje como *Adaline*. O método utilizado pelos dois para o treinamento do *Adaline* é o conhecido algoritmo dos

mínimos quadrados (LMS - Least Minimum Squares), no qual o ajuste dos valores das conexões, chamadas de "pesos", visava a minimização de uma função de erro dada pela soma dos erros quadrados obtidos para cada um dos padrões a serem aprendidos.

Nesta época ainda, a Inteligência Artificial apresentou duas tendências bem definidas. A primeira, baseada no processamento seqüencial, constituiu a base dos sistemas de computação em tempo real da época e da chamada inteligência artificial tradicional. Dentro desta tendência, conhecida como Processamento Digital Clássico (CDP), o sistema de computação é visto como uma mecanização seqüencial de processos ou eventos.

A segunda tendência, baseada no Processamento Paralelo Distribuído (PDP) e nas Redes Neuronais, surgiu como tentativa de imitar as estruturas neuronais biológicas. Através dos sistemas paralelos de processamento, a computação e a inteligência artificial são caracterizadas pela resposta coletiva a uma entrada. Nesta tendência, a informação contida no sistema é representada por padrões de ativação de elementos semelhantes ao neurônio biológico, elemento básico de processamento do cérebro humano.

Ao contrário dos sistemas CDP, o processamento paralelo envolve operações simples executadas por pequenos elèmentos processadores. Cada elemento executa uma função básica simples independente dos outros elementos e em paralelo com eles. Tais elementos são interligados em forma de rede de acordo com uma determinada topologia.

2.1.2 - Queda do interesse

Todas duas tendências aparentavam ser viáveis e áreas de pesquisa igualmente prometedoras. No entanto, no início dos anos 60, vários fatores significativos contribuíram para o aumento do interesse e importância dos sistemas de processamento seqüencial, deixando os sistemas de processamento paralelo em segundo plano.

Entre os fatores que contribuiram para a perda de interesse pelos sistemas paralelos, entre eles as redes neuronais, temos :

- o sucesso na implementação de arquiteturas de computação von Neumann;

- os rápidos avanços de hardware e software de apoio ao processamento seqüencial;

- a limitada capacidade multiprocessadora da época, necessária em sistemas paralelos;

- o escasso conhecimento das características biológicas fundamentais do sistema neuronal, o que atrasou as pesquisas na área técnica correspondente;

- as conclusões, consideradas por muitos como devastadoras, sobre o potencial das redes neuronais que chegaram Minsky e Papert em seu livro "Perceptrons" (1969). Suas conclusões causaram um impacto negativo nas pesquisas.

A publicação do livro "Perceptrons", junto com a rápida e bem sucedida expansão dos sistemas de computação von Neumann resultaram na escassez de recursos direcionados ao estudo das redes neuronais artificiais. Assim, os estudos foram concentrados em pequenos grupos de pesquisa, dos quais os dois mais ativos eram dirigidos pelo professor Frank Rosenblatt e pelo professor Bernard Widrow.

Minsky e Papert são culpados de desencorajar as pesquisas sobre aprendizado de máquinas em rede, interrompendo esta promissora linha de trabalho. Devemos ressaltar que eles apenas mostraram as limitações de certo tipo de redes, mais simples, em tarefas de reconhecimento de padrões, sendo que, as conclusões às quais chegaram não poderiam ser generalizadas a redes mais complexas, como a rede multicamadas, por exemplo. Um outro detalhe importante é que nesta época não existiam algoritmos de treinamento confiáveis para tais estruturas como existem hoje.

2.1.3 - Novo auge

Dado que os sistemas tradicionais abrangeram inúmeras áreas de aplicação, muitas dificuldades inerentes ao processamento seqüencial tornaram-se evidentes. Aplicações industriais relacionadas com o processamento de conhecimento, robótica, classificação em padrões, síntese e reconhecimento de voz e visão, encontraram problemas que sobrecarregavam o processamento simbólico dos programas da corrente tradicional.

Foi assim como, novas teorias surgidas no início e meados dos anos 80 junto com a potencialidade de novos modelos neuronais e algoritmos de treinamento levaram a uma retomada do interesse pelas redes neuronais.

Outros fatores também influenciaram nessa retomada, entre eles :

- a constante procura por sistemas de processamento paralelo com velocidade e desempenho elevados;

- os avanços significativos na microeletrônica e nas tecnologias de hardware associadas;

- avanços significativos na área biológica;

- publicações que atraíram a atenção de cientistas e engenheiros, principalmente "Neural Networks and Physical Systems with Emergent Collective Computational Abilities" de Hopfield (1982) e "Parallel Distributed Processing" de Rumelhart e McClelland (1988).

A prova deste auge foi mostrada na Primeira Conferência Internacional da IEEE sobre Redes Neuronais em 1987, além de duas conferências anuais nos dois anos seguintes (88 e 89).

2.2 - DESCRIÇÃO

As Redes Neuronais Artificiais, ou simplesmente, Redes Neuronais, constituem modelos de processamento distribuído paralelo. Elas consistem de um conjunto de processadores simples conectados entre si que operam em paralelo e possuem a capacidade de mudar o comportamento através de treinamento dinâmico.

Os elementos processadores da rede são conectados através de ligações de valor variável, chamadas comumente de pesos. Os valores das ligações determinam o comportamento global da rede. Tais valores são ajustados visando uma determinada relação entrada-saída desejada.

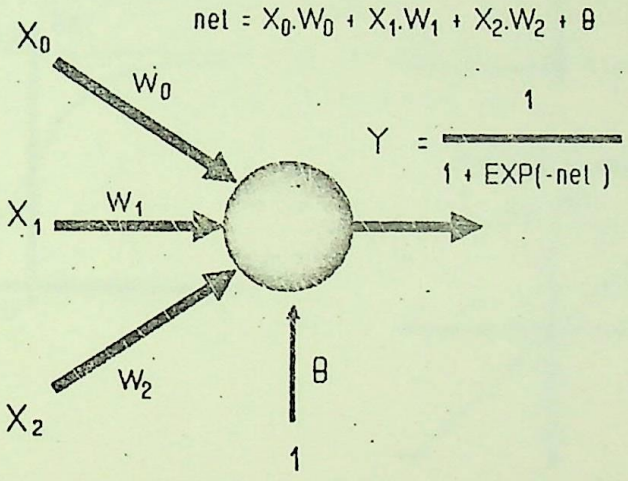


Figura 2.1 - Elemento básico da rede neuronal.

O elemento básico da rede, o neurônio, é mostrado na figura 2.1. A soma das entradas, multiplicadas pelos respectivos pesos w_i, é passada através de uma função normalmente não linear chamada de *função de ativação*. O neurônio é caracterizado também por um valor limite interno θ, conhecido como "*threshold*" ou "*offset*". A entrada do neurônio é dada pela seguinte equação:

$$net = \sum_i w_i . x_i + \theta \qquad (2.1)$$

e a saída do neurônio, pela equação :

$$y = f(net) = f(\sum_i w_i . x_i + \theta) \qquad (2.2)$$

A função de ativação f(t) é geralmente uma função não-linear limitada. Na figura 2.2 aparecem algumas das funções de ativação mais utilizadas. A escolha depende da aplicação a ser desenvolvida.

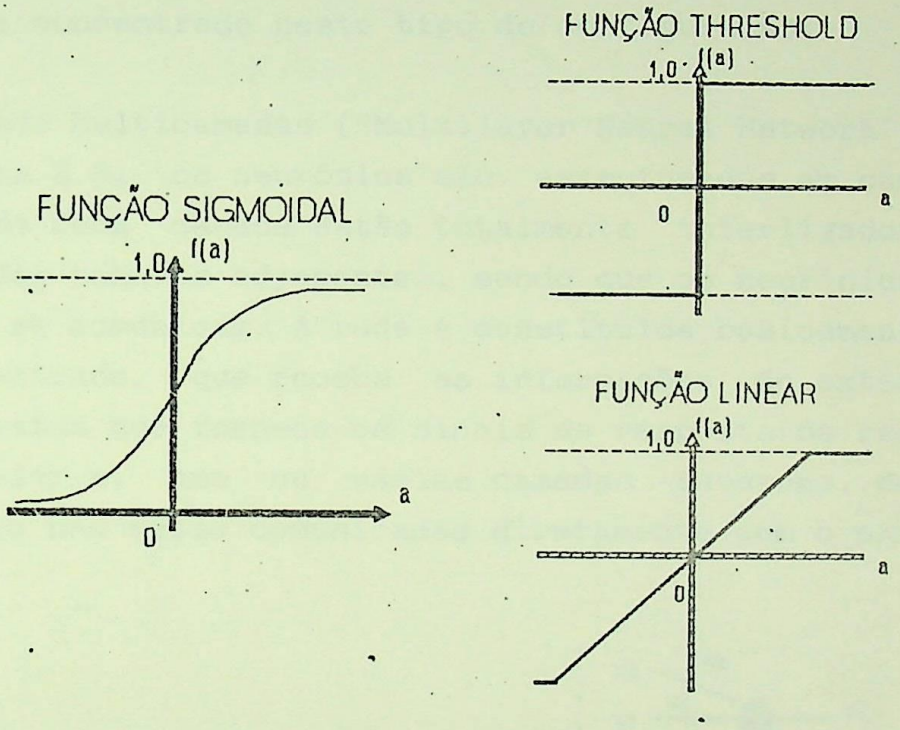


Figura 2.2 - Funções de ativação mais comuns.

A rede neuronal é definida pela estrutura formada pelas interconexões, pelo tipo de não-linearidade utilizada e pelo método de treinamento empregado. O método de treinamento define um conjunto inicial de pesos e como estes devem ser ajustados.

Em [14], são mostradas algumas configurações de redes neuronais já conhecidas, junto com as equações que definem o funcionamento matemático e as regras de ajuste dos pesos,

Na área de controle, especificamente, a rede Multicamadas tem atraído muito a atenção dos pesquisadores, visto que tem

mostrado um bom desempenho na modelagem de funções tanto lineares como não-lineares e conta com um algoritmo de treinamento bem definido e de fácil implementação.

2.3 - REDE MULTICAMADAS

Devido a seu potencial, já mostrado em trabalhos publicados nas revistas e publicações mais reconhecidas da área, nosso estudo será concentrado neste tipo de configuração.

Na Rede Multicamadas ("Multilayer Neural Network"), mostrada na figura 2.3, os neurônios são estruturados em camadas. Os neurônios de cada camada estão totalmente interligados com os neurônios das camadas adjacentes, sendo que os neurônios da mesma camada não se comunicam. A rede é constituída basicamente por uma camada de entrada, que recebe as informações do exterior, uma camada de saída que fornece os sinais de resposta da rede para o mundo exterior e, uma ou várias camadas internas, denominadas assim porque não estão comunicadas diretamente com o exterior.

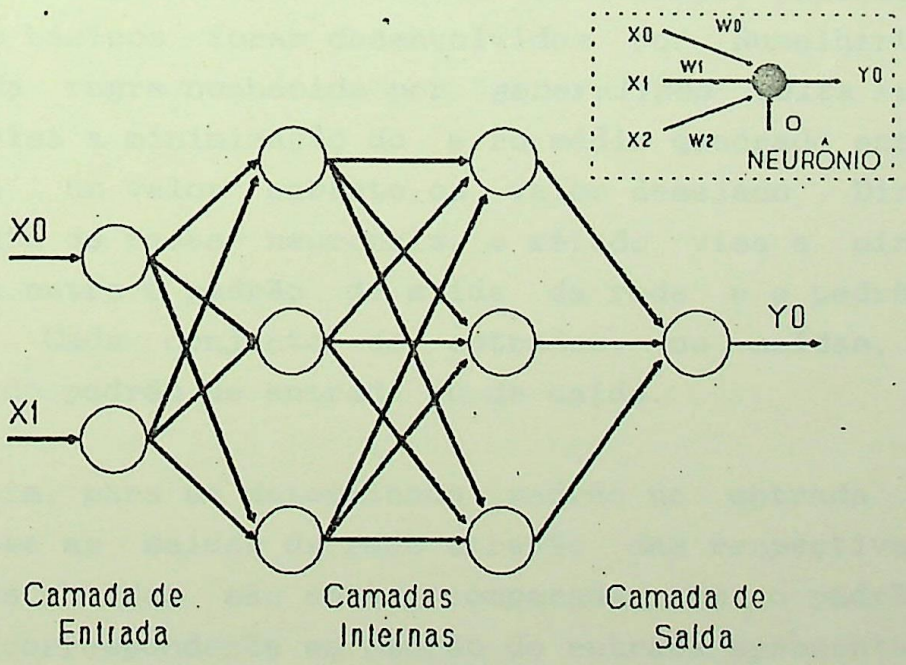


Figura 2.3 - Rede multicamadas.

Os neurônios internos; ou seja, aqueles que pertencem às camadas internas, são necessários, basicamente para a modelagem de funções não-lineares, e servem como supressores de ruído, extraindo pequenos desvios nas entradas [15].

As saídas são calculadas camada por camada no sentido entrada->saída, utilizando as saídas da camada anterior como entradas dos neurônios da próxima camada.

O ajuste dos pesos da rede multicamadas tem sido realizado com sucesso através do algoritmo de treinamento *Back-propagation*, ou Algoritmo de Propagação Reversa, o qual não é mais do que uma generalização do algoritmo dos mínimos quadrados.

2.4 - ALGORITMO DE TREINAMENTO "BACK-PROPAGATION"

Vários algoritmos tem sido criados para o treinamento da rede multicamadas, entretanto a maioria deles são versões modificadas ou incrementadas do algoritmo de propagação reversa.

Este algoritmo constitui um método iterativo cujos conceitos básicos foram desenvolvidos por Rumelhart em 1986, através da regra conhecida por "*generalized delta rule*" [16]. O método visa a minimização do erro médio quadrado entre um valor obtido e um valor correto ou valor desejado. Direcionado ao treinamento de redes neuronais, o método visa a minimização da diferença entre o padrão de saída da rede e o padrão de saída desejado. Cada conjunto de entradas, ou saídas, define um determinado padrão de entrada ou de saída.

Assim, para um determinado padrão de entrada disponível, calculam-se as saídas da rede através das respectivas equações. As saídas obtidas são então comparadas com o padrão de saída desejado correspondente ao padrão de entrada apresentado. O valor do erro definido pelos dois padrões de saída é propagado através da rede no sentido inverso ao de funcionamento normal, utilizando para isto equações bem definidas, que são a base do algoritmo.

Com a propagação reversa do erro obtêm-se a variação necessária em cada peso w da rede visando a diminuição do erro apresentado. Com a apresentação contínua dos vários padrões de treinamento disponíveis o método tende a minimizar o erro médio total (ou global), isto é, o erro médio definido para todo o conjunto de padrões, calculado a partir dos valores de erro obtidos para cada padrão.

Em termos de diagrama de blocos, o algoritmo pode ser representado pela figura 2.4.

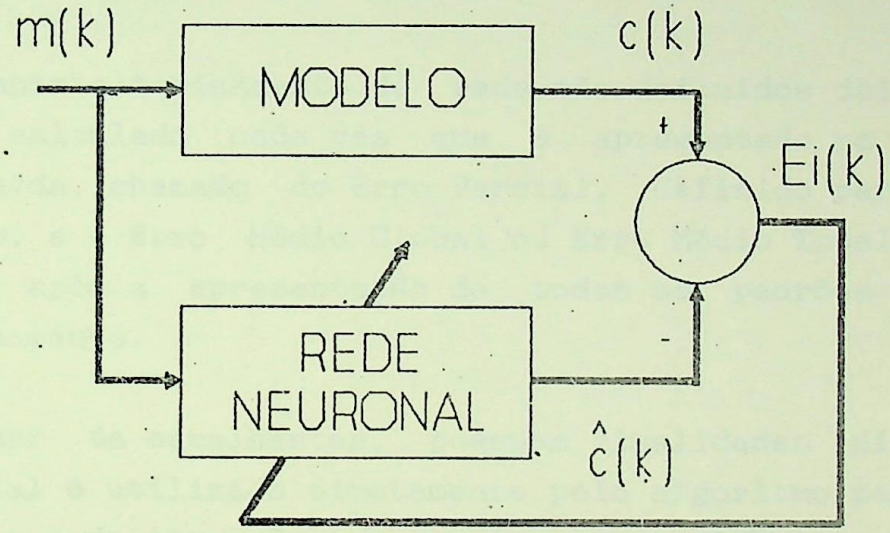


Figura 2.4 - Diagrama de blocos do algoritmo *back-propagation*.

2.4.1 - Função de Ativação

Como mostraremos mais tarde, o algoritmo *back-propagation* requer que a função de ativação utilizada pela rede seja contínua e diferenciável. Tal é o caso da função sigmoidal (figura 2.2):

$$f(x) = \frac{1}{1 + e^{-x}} \qquad (2.3)$$

A função sigmoidal é caracterizada por uma região em torno de x=0 na qual a saída é aproximadamente linear. Fora desta

região, a saída satura à esquerda em um valor mínimo e à direita em um valor máximo. Para o caso da equação (2.3) estes valores são respectivamente 0 e 1. Deve-se observar agora que, os valores de entrada e saída do neurônio, e portanto da rede em si, deverão ser limitados entre 0 e 1, daí portanto a necessidade de uma normalização nos valores a serem manuseados pela rede. Tal normalização será feita então, utilizando a faixa de valores esperada para cada uma das entrada e para cada uma das saídas.

2.4.2 - Erro parcial e erro global

Durante o treinamento da rede são definidos dois tipos de erro, um calculado cada vez que é apresentado um padrão de entrada-saída, chamado de Erro Parcial, definido para cada uma das saídas, e o Erro Médio Global ou Erro Médio Total, o qual é calculado após a apresentação de todos os padrões disponíveis para treinamento.

Apesar de semelhantes, possuem finalidades diferentes. O erro parcial é utilizado diretamente pelo algoritmo para o ajuste dos pesos a cada iteração, enquanto o erro global corresponde à soma dos erros parciais de todo o conjunto de treinamento, sendo a função a ser minimizada pelo algoritmo. O erro global pode ser utilizado para acompanhar o desempenho do treinamento, e com isto, determinar o fim do mesmo.

O erro parcial é definido pela equação :

$$E = \frac{1}{2} \sum_{j} (d_j - y_j)^2 \qquad (2.4)$$

onde j representa o somatório sob os neurônios da camada de saída, y_j é a saída atual do neurônio de saída j, e d_j é a saída desejada para o mesmo neurônio correspondente ao padrão de entrada apresentado.

A função a ser minimizada, o erro médio global, é dado por:

$$E_t = \frac{1}{2} \sum_p \sum_j (d_{pj} - y_{pj})^2 \qquad (2.5)$$

onde p representa o somatório sob o número de padrões usados para o treinamento.

2.4.3 - Desenvolvimento matemático

Como já foi destacado, o ajuste dos pesos tem como objetivo a minimização da função de erro da equação (2.5). Tal ajuste é feito, segundo o algoritmo, proporcionalmente à derivada deste erro em relação a cada peso, isto é :

$$\Delta w_{ji} = -\eta \frac{dE}{dw_{ji}} \qquad (2.6)$$

onde j e i representam os índices dos neurônios ligados através do peso w_{ji} (figura 2.5). O parâmetro η é conhecido como taxa de aprendizagem.

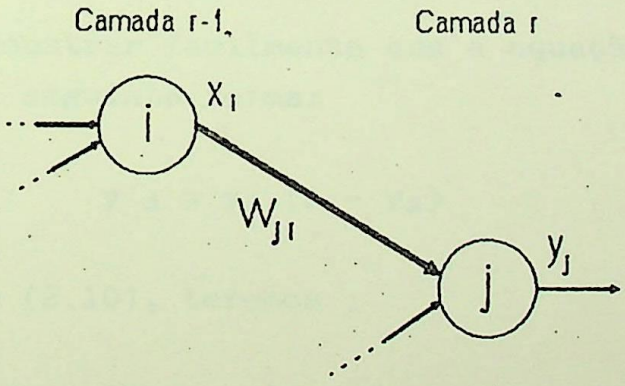


Figura 2.5 - Conexão entre dois neurônios.

A derivada na equação (2.6) é calculada utilizando o princípio da regra da cadeia, ou seja :

$$\Delta w_{ji} = -\eta \frac{dE}{dnet_j} \cdot \frac{dnet_j}{dw_{ji}} \qquad (2.7)$$

Da equação (2.2), para facilitar a análise, o *threshold* é tratado como um peso adicional, considerando que este é multiplicado por uma entrada unitária (figura 2.1). Assim, teremos então que:

$$\frac{dnet_j}{dw_{ji}} = \frac{d[\sum w_{ji}.x_i]}{dw_{ji}} = y_i \qquad (2.8)$$

A outra derivada da equação (2.7) será representada pelo parâmetro δ_j, deixando-a da seguinte forma:

$$\Delta w_{ji} = -\eta \cdot \delta_j \cdot y_i \qquad (2.9)$$

O valor de δ_j é obtido a partir de :

$$\delta_j = -(d_j - y_j) \cdot f'(net_j) \qquad (2.10)$$

A derivada $f'(net_j)$ é obtida utilizando a função de ativação adotada, neste caso, a função sigmoidal (equação 2.3) :

$$y_j = \frac{1}{1 + e^{-net_j}} \longrightarrow y'_j = \frac{e^{-net_j}}{(1 + e^{-net_j})^2} \qquad (2.11)$$

Pode-se demonstrar facilmente que a equação (2.11) pode ser escrita também da seguinte forma:

$$y'_j = y_j.(1 - y_j) \qquad (2.12)$$

Levando (2.12) em (2.10), teremos :

$$\delta_j = -(d_j - y_j) \cdot y_j \cdot (1 - y_j) \qquad (2.13)$$

Mas, o resultado obtido em (2.13) serve apenas para o cálculo de variações dos pesos conectados aos neurônios da camada de saída.

No entanto, para os pesos, associados às camadas internas, as variações podem ser obtidas através de um raciocínio semelhante.

Assim, para as camadas internas teremos :

$$\Delta w_{ji} = -\eta \frac{dE}{dw_{ji}} = -\eta \frac{dE}{dnet_j} \cdot \frac{dnet_j}{dw_{ji}}$$

$$\Delta w_{ji} = -\eta \frac{dE}{dnet_j} \cdot y_i = \eta \left(-\frac{dE}{dy_j} \cdot \frac{dy_j}{dnet_j}\right) \cdot y_i$$

$$\Delta w_{ji} = \eta \left(-\frac{dE}{dy_j}\right) \cdot f'_j(net_j) \cdot y_j \qquad (2.14)$$

E, novamente, a derivada parcial em (2.14) é representada pelo parâmetro δ_j, então :

$$\Delta w_{ji} = \eta \cdot \delta_j \cdot y_i \qquad (2.15)$$

O valor de δ_j pode ser obtido, segundo [17], em termos de quantidades conhecidas e/ou que podem ser calculadas. Sendo assim, temos :

$$-\frac{dE}{dy_j} = -\sum_k \frac{dE}{dnet_k} \cdot \frac{dnet_k}{dy_j}$$

$$= \sum_k \left(-\frac{dE}{dnet_k}\right) \cdot \frac{d\left[\sum w_{km} \cdot y_m\right]}{dy_j}$$

$$= \sum_k \left(-\frac{dE}{dnet_k}\right) \cdot w_{kj} = \sum_k \delta_k \cdot w_{kj} \qquad (2.16)$$

Levando (2.16) em (2.15), obtêm-se que :

$$\delta_j = y_j (1 - y_j) \sum_k \delta_k \cdot w_k \qquad (2.17)$$

onde k representa o somatório sob os neurônios da camada posterior à camada do neurônio j.

2.4.4 - Normalização

Dado que a função de ativação sigmoidal (figura 2.2 - equação 2.3) fornece um valor numérico limitado entre 0 e 1, logo, os cálculos derivados do algoritmo de treinamento são limitados também, entre 0 e 1. Sendo assim, deve ser preestabelecida uma faixa de variação para os parâmetros de entrada e de saída da rede. Não estamos nos referenciando aos valores precisos, mas sim de uma faixa de variação esperada, de acordo com as características e limitações físicas do modelo em estudo.

A partir desta faixa, determinada para cada uma das entradas e saídas, é feita uma normalização dos valores atuantes na rede, assim como também das saídas fornecidas pela mesma.

Os valores normalizados são determinados a partir da equação :

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (2.18)$$

na qual x_{max} representa o valor máximo e x_{min} o valor mínimo da faixa estabelecida. Desta forma, teremos que $x_n = 1$ para $x = x_{max}$, e que $x_n = 0$ para $x = x_{min}$.

Deve-se observar entretanto, que a saída da função sigmoidal dada pela equação (2.3) atinge os limites 0 e 1 para valores de entrada infinitamente grandes, positivos ou negativos. Isto significa que os limites da normalização dificilmente serão atingidos. Tal situação pode ser contornada de duas maneiras. A primeira, estaria em fixar limites bem acima e bem abaixo do esperado, no entanto o problema seria em determinar quanto acima ou quanto abaixo deverá se extrapolar.

Para evitar esta dúvida, a segunda opção fornece uma saída direta para este problema. A normalização pode ser modificada afim de restringir a faixa de atuação da equação (2.18) entre os valores 0,1 e 0,9. Isto pode ser conseguido com a seguinte modificação:

$$x_n = \left(\frac{0,9 - 0,1}{x_{max} - x_{min}} \right) \cdot (x - x_{min}) + 0,1 \qquad (2.19)$$

2.4.5 - Passos de treinamento

O algoritmo *back-propagation* é descrito, em resumo, pelos seguintes passos:

1) Inicializam-se os pesos com valores aleatórios pequenos, costuma-se muito especificar os pesos no intervalo entre -0,5 e +0,5.
É importante que os valores iniciais sejam distribuídos aleatoriamente para garantir a capacidade de aprendizagem.

2) Apresenta-se um dos padrões de entrada-saída disponíveis para o treinamento.

3) Efetuam-se os cálculos paralelamente através das camadas utilizando os valores atuais dos pesos, para obtenção das saídas da rede.

4) As saídas obtidas são comparadas com os respectivos valores de saída esperados definidos para o padrão de entrada utilizado, e calcula-se o erro de saída através da equação (2.4). Este erro é então propagado de forma reversa através das camadas internas usando as equações (2.13) e (2.17), calculando um valor de erro equivalente para cada um dos pesos que compõem a rede.

5) Através do erro correspondente, cada um dos pesos é ajus-

tado de acordo com a equação :

$$w_{ji}(k) = w_{ji}(k-1) + \Delta w_{ji}(k-1) \qquad (2.20)$$

onde o índice k refere-se ao instante de amostragem, logo, (k-1) refere-se ao instante ou iteração anterior. A parcela Δw_{ji} é obtida através dos valores de erro equivalentes obtidos no passo 4). Quanto maior o valor da taxa de treinamento η, maior será a variação nos valores dos pesos. Na prática escolhe-se um valor o mais alto possível que não provoque grandes oscilações nas variações.

Para aumentar a velocidade de convergência do treinamento, sem induzir oscilações, a equação (2.18) pode ser modificada afim de incluir um termo de momento ("*momentum term*"). Tal modificação é colocada em prática somando uma fração da variação anterior à ultima variação calculada para o peso em questão atual. Na equação, o termo é colocado da seguinte forma :

$$w_{ji}(k) = w_{ji}(k-1) + \Delta w_{ji}(k-1) + \alpha \cdot \Delta w_{ji}(k-2) \qquad (2.21)$$

onde α é basicamente uma constante, que determina o efeito da variação anterior na variação atual. O termo de momento procura manter as variações mais ou menos na mesma direção.

6) Apresenta-se o próximo padrão de entrada-saída, para o qual repetem-se os passos 3 a 5. O conjunto de padrões é apresentado de forma cíclica e contínua até que a saída acompanhe os resultados esperados. Se o conjunto de treinamento for adequadamente escolhido, a rede será capaz de generalizar o conhecimento adquirido e estendê-lo a situações diferentes das encontradas durante o treinamento. O comportamento do treinamento, assim como o fim do mesmo, é acompanhado através do cálculo do erro médio parcial E e do erro médio total Et.

O fim do treinamento pode ocorrer quando o valor do erro Et atinge um mínimo especificado previamente.

Através da variação do erro médio total pode-se ter uma idéia do comportamento do treinamento, isto é, se realmente a rede esta armazenando conhecimento e uma noção sobre a velocidade do mesmo. Por exemplo, oscilações no valor do erro podem ser um indício de valores muito altos da taxa de treinamento η ou da constante de momento α. Comportamento crescente ou invariante do erro são indícios de falhas nas constantes de treinamento, na normalização ou no conjunto de padrões. O acompanhamento do erro total pode, portanto, evitar perda de tempo com treinamentos inadequados ou mal projetados.

2.4.6 - Deficiências

Apesar da rede multicamadas ser uma das mais difundidas, o excessivo tempo de treinamento requerido pelo algoritmo é ainda um obstáculo nas aplicações onde o tempo desempenha um papel fundamental. Tal é o caso das aplicações em tempo real.

O tempo de convergência do algoritmo, dependendo do tamanho da rede pode atingir valores elevados, e este aumenta exponencialmente com o aumento do número de neurônios. No entanto, a procura por técnicas que minimizem o tempo de treinamento são os objetivos de muitas pesquisas de hoje no mundo inteiro [18,19].

Além do tempo, dois outros fatores merecem atenção especial. O primeiro é inerente aos procedimentos do tipo "*gradient descent*". Neste tipo de filosofia, a função a ser minimizada possui um mínimo principal conhecido como solução global e um ou vários mínimos parciais conhecidos como soluções locais. Pode ocorrer que, durante o processo de aprendizado, uma solução coincida com algum dos mínimos locais e resista a sair, o que por conseqüência nos levaria a uma solução que não é a melhor. Estudos para evitar este tipo de problema também são objeto de pesquisas [20,21]. Em [21], por exemplo, são propostas algumas condições sobre a arquitetura da rede e o ambiente de treinamento para garantir a convergência do *back-propagation*.

O segundo problema consiste no acerto da escolha dos parâmetros de treinamento, que como poderá ser sentido, é de vital importância para obter resultados amplamente satisfatórios.

Os efeitos de cada um destes parâmetros serão estudados com o objetivo de estabelecer algumas regras práticas para proceder à escolha certa nas aplicações de controle.

2.5 - ÁREAS DE ATUAÇÃO

Várias são as características das redes neuronais que chamam a atenção, principalmente na procura de soluções nos problemas nos quais as técnicas tradicionais não levaram a resultados aceitáveis.

Primeiramente, o paralelismo, que permite à rede analisar vários parâmetros simultaneamente e realizar, através de saídas independentes, vários tipos de controle, reduzindo com isto o tempo de computação necessário para tal fim.

A capacidade de aprender através de exemplos, permite que a rede seja treinada para representar um determinado modelo de comportamento. O treinamento pode ser efetuado tanto "*off-line*" como "*on-line*". O conhecimento adquirido pela rede é armazenado de forma distribuída, isto é, cada elemento de processamento contribui com uma pequena parte no cálculo da saída final. Tal característica dá à rede uma certa tolerância à falhas, visto que, um erro em qualquer uma das entradas não necessariamente comprometerá o comportamento global da rede [22].

A capacidade de aprendizagem da rede tem sido estendida tanto à modelagem de funções lineares como não-lineares, assim como em aplicações aleatórias, tais como o reconhecimento de voz e de escrita e a classificação em padrões.

A própria estrutura e mecanismo de funcionamento paralelo distribuído da rede cria uma imunidade considerável à presença de

ruído nos sinais de entrada [20].

Entre as aplicações já registradas que tem mostrado sucesso utilizando as características e vantagens acima relacionadas temos :

- Projeto de filtros adaptativos [23];
- Modelagem e controle de processos dinâmicos [24];
- Reconhecimento de voz e escrita para aplicação em robótica;
- Modelagem de processos químicos [25];
- Modelagem e otimização de processos [19];
- Processamento de sinais [26];
- Previsão de carga elétrica [27];
- Diagnóstico de falhas [28];

e muitas outras que ainda estão em desenvolvimento.

2.6 - IMPLEMENTAÇÃO PRÁTICA

A implementação da rede refere-se à escolha do ambiente de desenvolvimento da mesma. Para isto devemos fazer um levantamento, a partir das fases de concepção e de projeto, sobre os requisitos de capacidade, velocidade e memória exigidos pelo sistema, para então escolher o ambiente e as ferramentas de desenvolvimento mais adequadas.

2.6.1 - Softwares aplicativos

A escolha de um software está ligado aos requisitos de hardware, os quais basicamente são relacionados à capacidade, velocidade, disponibilidade de softwares e acesso aos dados. As *workstations* e PC´s são os ambientes de desenvolvimento para redes neuronais mais populares, no entanto, os *mainframes*, supercomputadores e os processadores paralelos são os mais indicados. Para a utilização em PC´s são disponíveis placas

aceleradoras para realizar cálculos numéricos com maior rapidez.

Já existe no mercado um grande número de softwares para o desenvolvimento de aplicações envolvendo redes neuronais. Tais aplicativos possuem a capacidade de realizar os seguintes passos:

- escolher uma determinada estrutura;
- determinar o tamanho da rede (número de camadas e neurônios);
- determinar os parâmetros de treinamento;
- selecionar entre vários tipos de apresentação;
- carregar um determinado conjunto de padrões para treinamento;
- inicializar a rede;
- especificar critérios de finalização;
- iniciar o treinamento;
- monitoração sob diversas maneiras;
- testes e avaliação.

2.6.2 - Implementações em hardware

Embora as pesquisas na área de implementação de redes neuronais ainda não tenham chegado a uma estrutura ótima, muitos são os esforços destinados ao estudo e teste de estruturas que podem ser criadas com a tecnologia disponível, visto que, nas aplicações em tempo real o paralelismo é de vital importância para realizar o maior número de cálculos matemáticos no menor tempo possível.

A simulação das redes neuronais requer recursos computacionais maciços. Os supercomputadores são poderosos o suficiente para realizar cálculos complexos, no entanto eles são caros e sua arquitetura não é a mais indicada.

São conhecidos vários trabalhos publicados, alguns dos quais são relacionados a seguir:

1) Em [29] são apresentados os resultados de testes de associação e reconhecimento de padrões e caracteres utilizando um circuito integrado de silício CMOS analógico, fabricado recentemente, com capacidade de treinamento digital. Foi desenvolvida uma interface e um ambiente para teste e verificação da funcionalidade e desempenho dos circuitos que emulam as unidades da rede.

2) A referência [30] apresenta uma discussão sobre a implementação analógica frente a digital, além de propor uma estrutura de neurônios denominada módulos lógicos programáveis dinamicamente (*dinamically programable logic modules*), os quais podem ser implementados através de demultiplexadores controlados digitalmente.

3) A referência [31] mostra uma análise sobre implementações baseadas em circuitos integrados VLSI disponíveis no mercado e tecnologia VLSI dedicada.

4) Em [32], examina-se a implementação de alguns modelos de redes neuronais em computadores com arquiteturas paralelas SIMD (*Single Instruction Multiple Data*), a qual executa a mesma instrução em cada um dos processadores ao mesmo tempo, utilizando seu próprio conjunto de dados, e MIMD (*Multiple Instruction Multiple Data*), onde cada processador roda seu próprio programa agindo sobre seus próprios dados.

Outras correntes prevêem ainda circuitos integrados dedicados mais poderosos ou processadores óticos.

CAPÍTULO 3

REDES NEURONAIS EM SISTEMAS DE CONTROLE

3.1 - REDES NEURONAIS NA MALHA DE CONTROLE

As técnicas de controle adaptativo tem sido desenvolvidas basicamente para processos que trabalhem sob condições inesperadas ou difíceis de serem previstas, e portanto, impossíveis de serem levadas em conta na construção de um modelo representativo.

Tais técnicas envolvem a construção de um modelo matemático que descreva a dinâmica do processo e o ajuste contínuo de um controlador baseado no modelo criado.

Mas, as técnicas adaptativas utilizadas comumente hoje em dia, tem falhado quando um modelo, com representatividade satisfatória, for difícil de ser obtido, ou mesmo, quando o modelo não leva em conta incertezas ou complexidades presentes no processo [33].

É precisamente diante deste tipo de deficiência que o sistema de controle neuronal tem mostrado um bom desempenho, principalmente, no manuseio de processos altamente não-lineares [34, 35, 36, 37, 38].

A utilização de redes neuronais na malha de controle tem ocorrido com vários objetivos. Entre os mais comuns estão a identificação e o controle de processos, funções que serão estudadas com detalhes neste trabalho. Outras aplicações, algumas já relacionadas no item 2.5 incluem, o processamento de sinais para controle e gerenciamento de alarmes, a detecção e correção

de falhas em sensores [39], reconhecimento de padrões em sistemas especialistas e no controle de funções relacionadas à robótica, tais como visão, reconhecimento de voz e controle de braços mecânicos [40, 33].

3.2 - IDENTIFICAÇÃO DE PROCESSOS

O treinamento de um emulador neuronal é similar ao procedimento de identificação de sistemas da teoria do controle tradicional. O procedimento consiste, a grosso modo, em determinar a estrutura e os parâmetros do sistema como função das saídas observadas. Para tal fim a relação entre os sinais de entrada e de saída é analisada para criar um modelo matemático.

Visto de uma outra forma, o objetivo da identificação é construir um modelo adequado que, quando submetido à mesma excitação, apresente na saída um comportamento semelhante ao do processo.

Em termos de redes neuronais, pretende-se treinar uma rede que, quando submetida à mesma entrada u(k) do processo, produz uma saída $\hat{c}(k)$ aproximadamente igual à saída real c(k), como mostrado na figura 3.1.

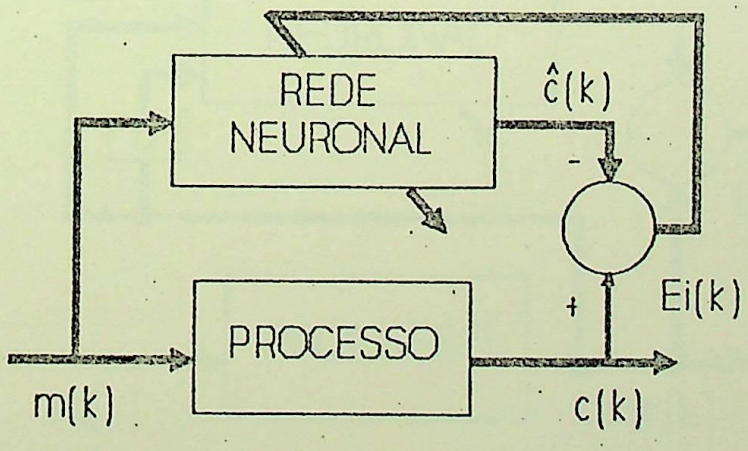


Figura 3.1 - Identificação de processos - Diagrama de blocos básico.

Pretende-se através da identificação fazer com que a rede imite o mapeamento entrada-saída apresentado pelo processo. No caso de um sistema de controle, a entrada representa o sinal de controle, fornecido por um operador ou um microprocessador, e a saída representa a variável sob controle.

No processo adotado para análise, o acionamento de um motor CC, o sinal de controle equivale à tensão de controle Vc que alimenta o circuito de disparo, enquanto a saída equivale à tensão gerada pelo tacogerador, indicando a velocidade angular da máquina.

O modelo identificado pela rede poderá facilmente ser utilizado no ajuste de um controlador, como mostraremos mais tarde, ou como parte de um esquema de otimização de processos industriais.

O esquema de identificação utilizando uma rede neuronal é mostrado mais especificamente na figura 3.2. O padrão de entrada da rede é formado a cada instante de amostragem pelo sinal de referência naquele instante junto com amostras anteriores do próprio sinal de referência e da saída amostrada do processo.

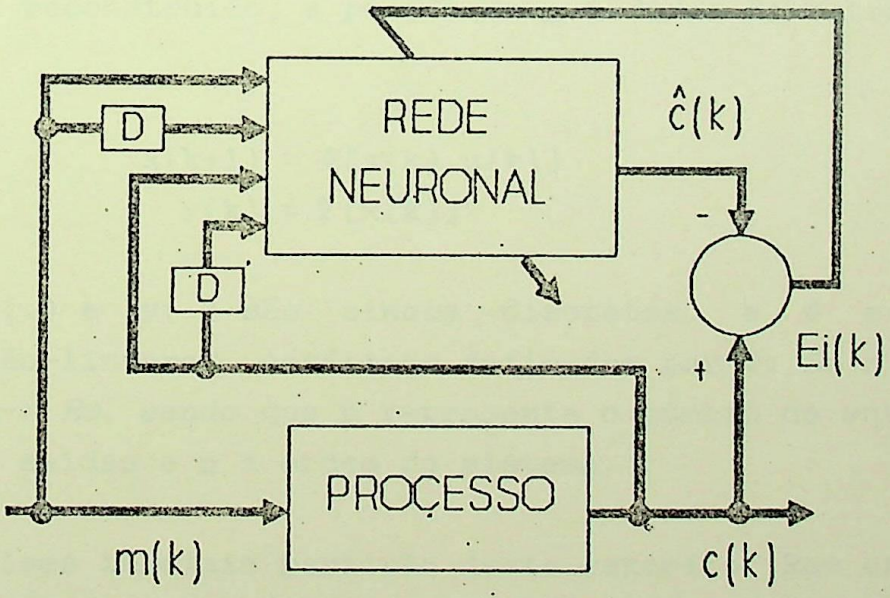


Figura 3.2 - Identificação - Esquema completo.

No capítulo 4 estudaremos os resultados da identificação e ajuste do controlador utilizando redes com números variados de entradas. Pretende-se com isto definir o número de amostras anteriores necessárias para obter um desempenho aceitável, além de conhecer como este número afeta tanto o desempenho como o tempo de computação envolvido, o qual constitui um dos principais obstáculos dos sistemas digitais.

Segundo a referência [35], a escolha da estrutura dos modelos a serem utilizados, tanto para identificação como para o controle de processos, representa um ponto importante na solução do problema. Para obter bons resultados devemos tomar como base, conceitos relacionados com o comportamento entrada-saída do sistema. Descreve-se, neste artigo, alguns métodos conhecidos para a especificação de estruturas de modelos utilizados na identificação e controle de processo lineares, assim como para o ajuste dos parâmetros relacionados. O mesmo assunto é abordado com relação aos processos não-lineares, onde a teoria disponível não oferece soluções imediatas.

Seguindo o raciocínio apresentado em [35], para um dado sistema do tipo uma entrada e uma saída, representado pelas equações (3.1), assume-se que o estado do sistema pode ser conhecido, ou reconstruído, a partir de n medidas da entrada e da saída.

$$x(k+1) = \Phi[x(k),u(k)]$$
$$y(k) = \Psi[x(k)] \qquad (3.1)$$

onde u(.), x(.) e y(.) são sinais discretos, e Φ e Ψ são mapeamentos não-lineares estáticos definidos por Φ: $Rn \times Rp \rightarrow Rn$ e Ψ: $Rn \rightarrow Rm$, sendo que p representa o número de entradas, m o número de saídas e n a ordem do sistema.

As análises iniciais partirão deste conceito. Por enquanto, a conclusão mais importante é que o modelo a ser adotado deve ter uma estrutura análoga a do processo, isto é, ter uma relação similar do número de entradas e saídas e da ordem do processo.

A cada instante de amostragem, formado o padrão de entrada da rede, obtém-se, através dos respectivos cálculos, a saída $\hat{c}(k)$ da rede, a qual será então comparada com a saída real do processo $c(k)$. A comparação é feita com o intuito de calcular um valor de erro entre ambos, sendo que para isto é utilizada a equação do erro parcial apresentada no item 2.4.2.

O erro calculado, representado por $E_i(k)$, corresponde portanto ao erro naquele instante, o qual é propagado reversamente através das camadas da rede para o ajuste respectivo dos pesos, de forma análoga à mostrada nos passos do algoritmo de treinamento do item 2.4.5 com ligeiras modificações apenas: no lugar de um conjunto de padrões de entrada-saída pré-definido temos agora amostras de sinais coletados do processo. Desta forma, o erro médio total (item 2.4.2) é calculado agora sob um determinado intervalo de tempo, que deverá ser especificado previamente [41]:

$$E_t = \frac{1}{2} \sum_k \sum_j [c_j(k) - \hat{c}_j(k)]^2 \qquad (3.2)$$

onde o índice k representa o somatório sob o intervalo de tempo.

O erro parcial é calculado da mesma maneira que no item 2.4.2, sendo que a cada iteração o erro parcial calculado contribui, mesmo que de forma pequena, com uma parcela pequena do conhecimento total que será adquirido pela rede até o final do treinamento.

Devemos deixar claro que o ajuste dos pesos é efetuado a cada instante de tempo, ou seja, a cada iteração, através do erro $E_i(k)$ correspondente a esse instante. O erro E_t é usado apenas para determinar o fim do treinamento.

Alguns aspectos devem ser levados em conta na escolha de um sinal de referência adequado para o treinamento:

- a faixa de atuação esperada do sinal de referência que

será utilizado durante o regime de operação normal;

- condições de operação reais;

- comportamento da referência durante o regime permanente;

- grande número de variações, que permitam à rede assimilar a resposta dinâmica a ser modelada.

Um sinal de treinamento construído com estas características deverá ter condições de gerar resultados satisfatórios.

3.3 - O CONTROLADOR NEURONAL

O sistema de controle adaptativo analisado no item 1.1.4, tem servido como base para os sistemas de controle neuronais, os quais tem-se enquadrado dentro de dois tipos de filosofia, o controle de tipo indireto e o controle de tipo direto.

Na filosofia de controle indireto, o ajuste do controlador depende de um estágio prévio de identificação. Enquanto que, nos sistemas de controle direto, o ajuste é realizado utilizando uma determinada estratégia de adaptação do controlador que depende diretamente das entradas e saídas do processo, sem a necessidade de um modelo.

As duas filosofias tem sido objeto de estudos, tanto teóricos como práticos, utilizando redes neuronais na implementação. No entanto, devido a algumas vantagens que serão apresentadas a seguir, para as análises que serão desenvolvidas neste trabalho utilizaremos apenas sistemas do tipo indireto.

Em ambas filosofias o papel da rede é praticamente o mesmo, processar o sinal de referência e a realimentação do sistema para gerar, de acordo com as condições de treinamento utilizadas, o sinal de controle adequado, tal que, a diferença entre os dois sinais de entrada seja mínima. Tais funções são ilustradas, em

termos de diagrama de blocos, na figura 3.3.

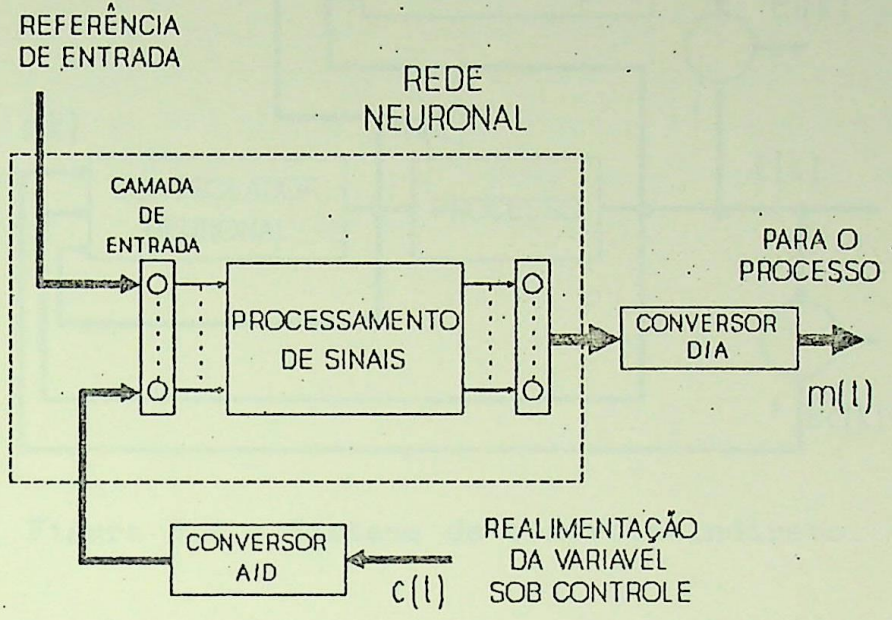


Figura 3.3 - Funções da rede na malha de controle.

3.3.1 - Controle indireto

O diagrama de blocos do sistema de controle adaptativo do tipo indireto utilizando redes neuronais pode ser visto na figura 3.4. O procedimento de identificação ocorre de forma "*on-line*", como nos sistemas de controle adaptativo tradicionais, para isto é utilizada uma rede neuronal cujo treinamento obedece ao procedimento do item 3.2. O modelo identificado é então utilizado para ajustar uma outra rede que fará o papel de controlador.

Como foi visto anteriormente, durante a fase de identificação, o erro utilizado para o ajuste dos pesos é a diferença entre a saída do processo e a saída da rede, que foi denominado erro de identificação (Ei). Para o ajuste do controlador agora, o erro a ser utilizado é dado pela diferença entre a saída do processo e a referência. Este erro é chamado de erro de controle, Ec(k).

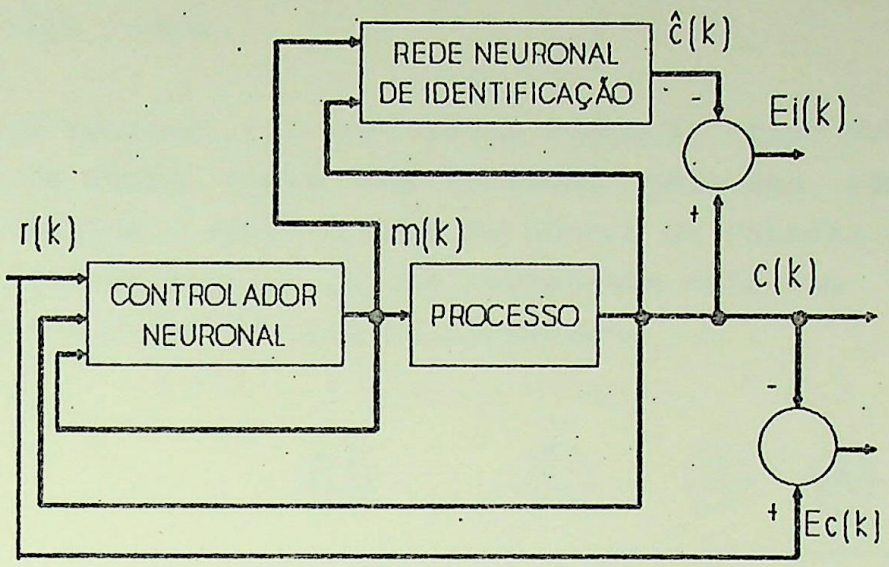


Figura 3.4 - Sistema de controle indireto.

Deve-se observar, no entanto que, segundo a teoria de redes neuronais apresentada no capítulo 2, para o ajuste dos pesos da rede controladora no esquema da figura 3.4 devemos utilizar o erro na saída do próprio controlador, e não o erro na saída do processo. Entretanto, não há como saber, de forma direta, o valor deste erro. É exatamente neste aspecto que se diferenciam as duas filosofias de controle introduzidas no início deste item, em como é tratado este problema. Veremos que o controle indireto apresenta uma vantagem e uma desvantagem diante do controle direto.

Em [16], Rumelhart e McClelland fazem as seguintes observações, nem todos os pesos de uma rede precisam ser variáveis, qualquer número de pesos podem ser fixos. Neste caso, o erro é propagado como antes, os pesos fixos simplesmente não são modificados.

Dado que o erro na saída pode ser facilmente conhecido a qualquer instante, e que, no controle indireto o esquema possui uma rede neuronal como modelo identificado do processo, podemos então propagar reversamente o erro na saída do processo Ec(k), através das camadas da rede de identificação "sem modificá-las" até obter na entrada desta, isto é, na saída do controlador, um

valor de erro equivalente que pode agora ser utilizado para o ajuste de seus pesos.

Isto é possível, já que várias redes do tipo multicamadas, colocadas em série, podem ser tratadas como uma rede só, cujo número de camadas é igual à soma do número de camadas de cada uma das redes individualmente, o que representa mais uma vantagem da utilização das redes do tipo multicamadas.

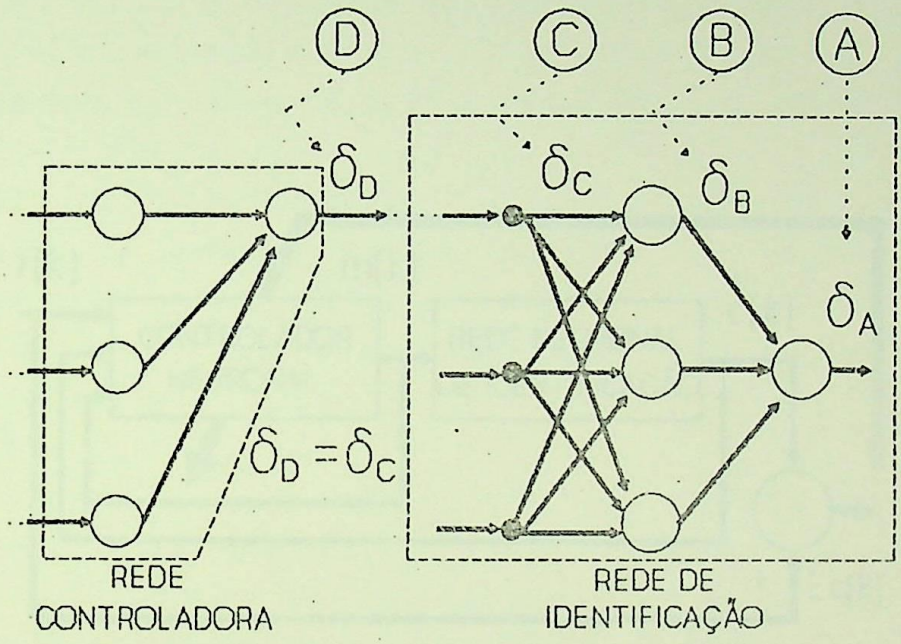


Figura 3.5 - Obtenção do erro equivalente para treinamento do controlador.

Este procedimento pode ser melhor ilustrado através da figura 3.5. Na camada de saída (ponto A), calcula-se o erro δ_A utilizado para o ajuste dos pesos ligados ao neurônio de saída da rede de identificação. Como trata-se de um neurônio de saída utilizamos a equação 2.13. Propagando reversamente δ_A podemos obter o erro no ponto B, δ_B, referente ao primeiro neurônio da camada interna, utilizando agora a equação 2.17. Da mesma forma pode-se calcular o erro δ_C com a mesma equação 2.17, ainda na rede de identificação. Tomando a rede controladora como uma extensão da rede de identificação, o erro δ_C corresponderia ao erro no ponto D, ou seja, $\delta_D=\delta_C$. Sendo assim, o erro δ_D pode ser

propagado pela rede controladora para ajustar seus pesos.

Um detalhe importante pode ser notado neste ponto, o ajuste do controlador não depende de qualquer parâmetro ou sinal vindo do processo, depende apenas do sinal de referência e realimentações de sinais das duas redes. Logo, o esquema real de ajuste do controlador, derivado do esquema da figura 3.4, pode ser mostrado então na figura 3.6. Mais tarde falaremos sobre o funcionamento do esquema da figura 3.4 durante a operação normal do sistema de controle.

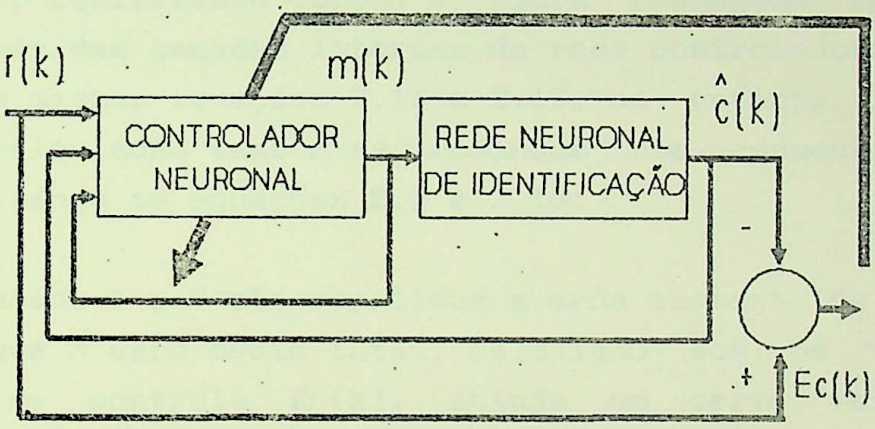


Figura 3.6 - Esquema de treinamento do controlador neuronal.

O ajuste da rede controladora obedece, então, aos seguintes passos:

1) Dado o sinal de referência no instante k, r(k), forma-se, junto com as realimentações tanto da saída da própria rede controladora como da rede de identificação (figura 3.6), o padrão de entrada da rede controladora;

2) A rede controladora, através dos cálculos pertinentes, fornece o sinal de controle m(k) que alimenta a rede de identificação, e esta por sua vez, no papel de modelo do processo, fornece a saída $\hat{c}(k)$ do sistema.

3) A saída c(k) é comparada com a sinal de referência para o cálculo do erro de controle Ec(k) utilizando a equação 2.4, lembrando apenas que temos uma só saída;

4) O erro Ec(k) é então propagado reversamente primeiro através da rede de identificação, utilizando as equações 2.13 e 2.17;

5) Considera-se o erro na saída do neurônio na camada de entrada da rede de identificação, que recebe o sinal de controle como sendo o erro equivalente na saída do neurônio da camada de saída da rede controladora (figura 3.5);

6) O erro equivalente obtido é agora propagado reversamente através das camadas internas da rede controladora utilizando as mesmas equações 2.13 e 2.17, no entanto, desta vez são calculadas também as mudanças nos respectivos pesos utilizando as equações 2.9 e 2.15;

7) Os passos 1 a 6 são repetidos a cada instante de amostragem até que o erro médio total, calculado sob os valores do erro de controle Ec(k), atinja um certo valor mínimo determinado, ou até que sejam completadas um número de iterações máximo, ambos valores determinados previamente.

Da mesma forma que o procedimento de identificação, este também é um processo dinâmico de ajuste. O somatório em k da equação 3.2 indica novamente um processo de soma de um determinado número de iterações.

O número de iterações utilizado para a atualização do erro médio total, o seu valor mínimo desejado e o número de iterações máximo, devem ser escolhidos adequadamente a partir de dados experimentais.

3.3.2 - Controle direto

No método de controle direto os parâmetros do controlador, neste caso, os pesos da rede, são ajustados de forma direta, a partir de um conhecimento qualitativo do processo.

No controle indireto, durante o ajuste do controlador, o processo, ou o modelo que o representaria, é tratado também como uma camada adicional invariável da rede controladora, só que agora o processo é representado por uma aproximação matemática simples. Procura-se através desta aproximação eliminar o tempo de computação necessário para o procedimento de identificação mostrado no controle indireto, no entanto, ao utilizarmos uma aproximação, pode-se reduzir, como conseqüência, a precisão dos resultados obtidos após o treinamento.

A escolha da filosofia adequada vai depender, basicamente, da complexidade do modelo a ser controlado e do tempo de computação disponível.

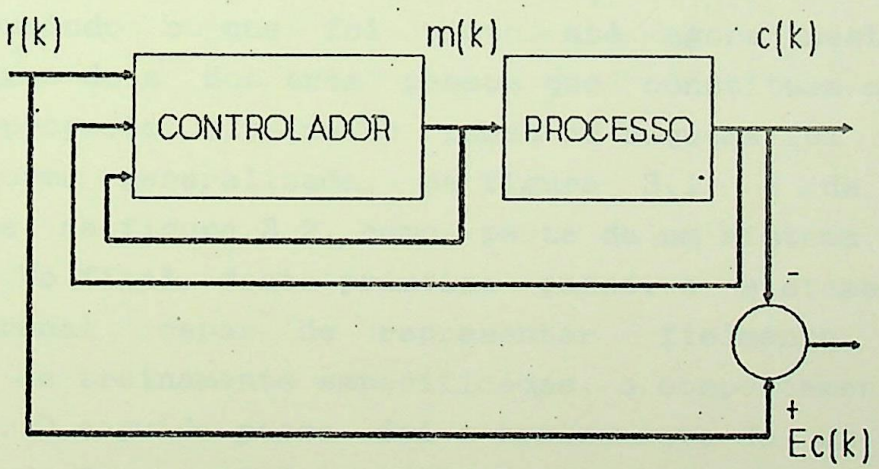


Figura 3.7 - Sistema de controle direto.

O diagrama de blocos típico de um sistema de controle direto é mostrado na figura 3.7, onde a rede faz o papel de controlador da malha e nota-se a ausência da malha de identificação.

Vários trabalhos tem sido desenvolvidos utilizando redes neuronais em uma estrutura de controle direto. Em [41], por exemplo, a planta é aproximada, para propósito de ajuste do controlador, utilizando um conhecimento prévio do seu comportamento, especificamente, sobre a maneira de como cada uma das variáveis de entrada influencia a saída. Desta forma, a influência de cada uma destas variáveis é determinada pela função "sign":

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

Em [42], a planta é aproximada, similarmente, de acordo com a resposta da saída frente a variações em cada uma das entradas.

3.4 - ESQUEMA DE CONTROLE EM REGIME DE OPERAÇÃO NORMAL

Resumindo o que foi visto até agora neste capítulo, apresentamos dois dos três passos que constituem o esquema de controle proposto. O primeiro passo ou esquema foi apresentado, na sua forma generalizada, na figura 3.1, e de forma mais específica, na figura 3.2, como parte de um sistema de controle indireto. No final deste primeiro passo, o sistema possui uma rede neuronal capaz de representar fielmente, dentro das condições de treinamento especificadas, o comportamento da planta em estudo. O segundo passo foi o treinamento de um controlador, mostrado na figura 3.6, utilizando o modelo identificado no primeiro passo.

Como em qualquer projeto de um sistema de controle, quanto maior o número de detalhes ou especificações conhecidos sobre o processo a ser controlado maior será a confiabilidade do sistema desenvolvido.

Chamamos a atenção no entanto de que no caso de redes

neuronais nos referimos a detalhes de ordem externa e não aos que podemos chamar de detalhes internos. Especificações internas são aquelas relacionadas diretamente com a modelagem do processo quando utilizamos a teoria de controle clássica, como por exemplo, a freqüência natural de oscilação, ganho, ordem do sistema, possíveis atrasos, constantes de tempo, etc., parâmetros que comumente são levantados para o projeto de malhas de controle. Já, quando nos referimos às especificações externas citamos parâmetros de ordem qualitativa, conhecidos quando tratamos a planta como uma "caixa preta", como por exemplo, as características e o comportamento dos sinais de entrada e saída, a faixa de atuação real destes sinais, o regime de funcionamento, e principalmente, a relação em termos de resposta entre as variáveis de entrada e as variáveis de saída, mais a nível qualitativo do que quantitativo, isto é, uma noção quanto à influência que as entradas exercem sobre as saídas.

Esta vantagem é talvez uma das mais importantes das redes neuronais, diminui-se cada vez mais a preocupação que envolve a definição e implementação de estruturas complexas para a modelagem de sistemas, visto que a rede neuronal aprende diretamente de exemplos, em termos de entradas e saídas, e quanto melhor definido o conjunto utilizado para treinamento, melhor será o resultado.

Dado que o ajuste do controlador não depende de informações geradas pelo processo pode-se supor que os dois esquemas apresentados, figura 3.2 e figura 3.6, poderiam trabalhar simultaneamente para o treinamento das redes através do esquema da figura 3.4, o que a primeira vista parece não possuir nenhum inconveniente. No entanto, devemos notar que, o ajuste do controlador depende da rede de identificação, ou melhor, do conhecimento adquirido por esta. Para perceber isto mais claramente devemos lembrar que o erro utilizado no ajuste do controlador é propagado através da rede de identificação primeiro, e, partimos do princípio de que esta representa fielmente o comportamento do processo. Caso isto não seja verdadeiro, quando o erro de controle for propagado reversamente ele carregará junto o erro de identificação, que poderá,

dependendo do caso, se sobrepor ao de controle, levando a variações erradas nos pesos, e como conseqüência, nenhum conhecimento verdadeiro será adquirido pelo controlador.

Desta forma, antes de iniciar o procedimento de ajuste do controlador deve-se garantir um grau de treinamento mínimo para a rede de identificação. Daí a importância de uma escolha certa do valor mínimo do erro médio total que determinará o fim do treinamento.

Treinado com sucesso, o controlador este deverá ser capaz de acionar corretamente o processo de acordo com a referência desejada.

O sistema passa a funcionar agora sob o terceiro esquema, para operação em regime normal. Este esquema já foi introduzido através da figura 3.4, quando citou-se sobre a estrutura dos sistemas de controle indireto.

O sistema funciona da seguinte forma: o controlador recebe o sinal de referência vindo do operador, com o qual gera na saída o sinal de controle adequado, de acordo com o padrão constituído na sua entrada. O sinal de controle alimenta a planta, que por sua vez gera a saída correspondente ao sinal de controle aplicado. Após o treinamento, esta saída deverá acompanhar sempre a referência aplicada.

Esta seqüência ocorre a cada instante de amostragem da mesma forma, realimentando os sinais indicados na figura 3.4 para formar os respectivos padrões de entrada das redes.

Para garantir um desempenho satisfatório, a saída c(k) do processo é comparada a cada instante com a referência r(k) que representou durante o treinamento do controlador o valor de saída desejado. A diferença entre estes dois sinais pode ser obtida em termos de porcentagem, para o qual especifica-se uma tolerância aceitável. Caso em algum momento o erro percentual calculado ultrapasse o limite estabelecido, o sistema deverá reagir afim de diminuir a diferença apresentada.

O maior problema do sistema é como este deverá reagir para corrigir desvios que possam ocorrer.

Muitos dos trabalhos desenvolvidos envolvendo esquemas de controle indireto com redes neuronais tem apresentado vários tipos de configurações para o sistema, assim como aspectos sobre como o treinamento é realizado. No entanto, nenhum deles apresenta conceitos ou técnicas sobre como garantir que o desempenho da rede se mantenha satisfatório durante o regime de operação normal ou, em outras palavras, como agir caso haja uma queda no desempenho do sistema.

Podemos observar que, a diferença entre r(k) e c(k) pode ser provocada por uma variação nos parâmetros originais do processo ou por influências do meio. Estas variações podem ser desconhecidas e/ou imprevisíveis.

Seja qual for o motivo de uma mudança de comportamento, deve-se definir uma nova rotina de ajuste para evitar que tais variações continuem prejudicando o comportamento da malha.

Independente da causa da variação, seja nos parâmetros do processo, incluindo os transdutores, ou no meio em que ele se encontra, devido possivelmente à interferência de outros equipamentos, não basta apenas ajustar a rede controladora, mas ajustar, primeiro, a rede de identificação, afim de que esta continue representando o comportamento do processo.

Sendo assim, para que a mudança seja assimilada pela malha na figura 3.4, o erro Ei(k) de identificação, definido pelos sinais c(k) do processo e c(k) da rede, deve ser calculado e propagado para o ajuste dos pesos. A seguir, o erro de controle Ec(k) é calculado e propagado reversamente primeiro através da rede de identificação para obtenção do erro equivalente, da mesma forma como foi mostrado no item 3.3.1, isto é, sem modificá-la agora, e depois, propaga-se o erro equivalente obtido através da rede controladora, procedendo com o respectivo ajuste de seus pesos.

Esta seqüência de ajuste deve ser realizada dentro do intervalo de amostragem, o qual deve ser adequadamente escolhido, segundo os conceitos que serão apresentados no item 4.1.4.

Reajustar apenas a rede controladora não seria correto, pois, como já mencionamos, o ajuste desta depende do fato de que o modelo identificado representa fielmente a dinâmica do processo. Se ocorrem mudanças no processo e o modelo não as acompanhar o ajuste do controlador será prejudicado.

Vale destacar que, com a rede neuronal implementada utilizando circuitos que exploram o seu paralelismo (ver item 2.6.2), o tempo de resposta cai bruscamente em relação aquele que utiliza um equipamento com processamento seqüencial, como é o caso dos microcomputadores ou microprocessadores. Com isto, a seqüência de ajuste anterior poderia ser efetuada sem problemas em sistemas com redes mais complexas ou, com redes simples, mais de uma vez dentro de um único intervalo de amostragem visando aumentar a velocidade de recuperação do sistema.

Neste ponto, cabe destacar novamente a influência das taxas de treinamento η e α, levando-se em consideração que as redes já passaram pelo devido treinamento. Para um novo reajuste de pesos utilizando o esquema da figura 3.4 é aconselhável que, pelo papel que desempenham no treinamento, as constantes sejam mantidas com valores pequenos para que não ocorram mudanças bruscas ou oscilações na saída, mas que permitam mudanças na rede.

Vários pesquisadores têm proposto a utilização de valores variáveis e não constantes para η e α, o que tem apresentado bons resultados. Em tais trabalhos, os parâmetros assumem dois valores definidos, no entanto, nós podemos estender esta variação para uma faixa de valores, sendo que o valor do parâmetro seria definido a cada instante do ajuste de acordo com o valor do erro detectado na saída.

Vale ressaltar novamente que somente através de uma análise mais profunda e variada das simulações, sob diversas condições de funcionamento, poderemos atingir resultados satisfatórios, além

de ser necessário definir algumas regras práticas que apoiem o projeto e implementação de outros sistemas de controle.

3.5 - METODOLOGIA DE DESENVOLVIMENTO DE APLICAÇÕES

Com o aumento do número de aplicações e a ampliação dos limites de atuação onde as redes neuronais tem mostrado resultados acima da expectativa, tornou-se necessário definir regras e conceitos que definam uma seqüência de desenvolvimento efetiva, visando garantir a confiabilidade e a precisão dos sistemas projetados.

A metodologia a ser adotada e seguida neste trabalho é constituída por quatro fases: definição, projeto, implementação e manutenção. Todas as fases, apesar de simples, são de suma importância. Análises superficiais podem levar, como em qualquer outro tipo de sistemas, a resultados insuficientes ou inaceitáveis.

3.5.1 - Definição do problema

Nesta primeira etapa pretende-se definir as caraterísticas do problema a ser resolvido, fixar os objetivos do sistema a ser projetado e estabelecer razões que justifiquem a utilização de redes neuronais na solução do problema.

As aplicações mais comuns que têm mostrado sucesso possuem algumas características em comum:

- dependem de vários parâmetros que interagem entre si;
- possuem facilidade de acesso a exemplos caracterizados em termos de entradas e saídas;
- envolvem processos difíceis de serem modelados adequada-

mente, devido à presença de comportamentos não-lineares ou interferências externas.

Alguns indícios podem mostrar que as redes constituem uma boa opção:

- as técnicas de computação convencional resultam insuficientes ou tornam as análises muito complexas;

- o problema requer raciocínios qualitativos e quantitativos complexos;

- a solução depende de parâmetros fortemente ligados entre si;

- dados ou sinais de fácil acesso, possívelmente contaminados por ruído ou propensos a erros;

- tempo de desenvolvimento disponível limitado.

Outra maneira, mais superficial, de descobrir a viabilidade de um sistema neuronal consiste em comparar as características e requisitos da aplicação com as qualidades mais destacadas das redes, o paralelismo, capacidade de aprendizagem e a generalização de conhecimento.

No capítulo 1 foi mostrada a estrutura e o funcionamento de um sistema de controle de velocidade de motor CC, assim como o comportamento de um sistema de controle adaptativo, no qual é baseado o sistema de controle neuronal. O objetivo do sistema é o controlar a velocidade do motor CC através de um sinal de referência que representa o padrão de comportamento desejado para a velocidade do motor, direcionado ao acionamento de um determinado processo ou máquina.

Com todas as informações disponíveis do processo, há duas questões que devem ser muito bem definidas. A primeira, consiste em determinar quais os parâmetros ou sinais relacionados direta ou indiretamente com o funcionamento do processo, seu papel e o

comportamento de cada um. A segunda questão refere-se à faixa de atuação de tais parâmetros, o que é de vital importância para o treinamento das redes.

Para completar esta fase, tendo já optado por uma solução a base de redes neuronais, deve-se especificar o objetivo específico da(s) rede(s), ou seja, a posição e o papel a ser desempenhado pela(s) mesma(s) dentro do sistema. Devemos especificar também requisitos referentes ao tempo e ao equipamento disponível, assim como o tipo e validade das fontes de dados e sinais necessários. E, finalmente, os resultados esperados.

3.5.2 - Projeto

3.5.2.1 - Caracteristicas dos neurônios

Como foi mostrado no item 2.2 o neurônio pode ter vários tipos de comportamento, definidos de acordo com a função de ativação empregada. Pode-se inclusive utilizar neurônios com funcões de ativação diferentes formando uma mesma rede. No entanto, a maioria dos trabalhos tem utilizado redes com apenas um tipo de função de ativação.

O tipo de saída que a rede deve fornecer é que determina a função que deve ser utilizada. O desenvolvimento matemático do item 2.4.3 foi feito baseado em neurônios com função de ativação sigmoidal.

3.5.2.2 - Estrutura e tamanho das redes

O projeto do sistema depende basicamente do tipo de rede a ser utilizada. Em [43], há um levantamento sobre as caracterís-ticas de algumas das redes mais utilizadas para orientar na

escolha da configuração mais adequada.

Neste ponto, torna-se necessário que optemos por um determinado tipo de rede para poder especificar mais os passos a seguir. O tipo de rede que tem mostrado melhores resultados em aplicações na área de controle é precisamente a Rede Multicamadas, introduzida no capítulo 2. Os passos de projeto serão realizados, portanto, utilizando esta configuração.

Resta agora definir o número e tamanho das camadas de cada uma das redes que serão utilizadas. O número de neurônios na camada de entrada é definido a partir do número de sinais ou informações diferentes que devem ser fornecidos à rede. Da mesma forma, o número de neurônios na camada de saída é definido pelo número de sinais ou informações que devem ser gerados pela rede.

Faltaria apenas especificar o número de camadas internas e o número de neurônios de cada uma delas. Várias são as referências nas quais tem-se comprovado que uma rede com apenas duas ou três camadas é capaz de aproximar qualquer mapeamento contínuo com uma precisão arbitrária, isto é, tomando cuidado apenas para que a rede tenha um número suficiente de neurônios internos para o armazenamento das informações [14, 44, 38]. Aumentar o tamanho da rede não é muito simples quando o tempo de processamento é limitado. Em geral, parte-se com um número mínimo de camadas e experimenta-se depois com redes maiores visando melhorar os resultados.

O número de neurônios mínimo deve ser tal que a rede evite cair nos chamados mínimos locais ("*local minima*"), risco que aumenta com um número de neurônios pequeno.

Segundo a referência [18], afirma-se que o algoritmo *back-propagation* apesar de aumentar o tempo de processamento de cada iteração com o aumento do número de neurônios, o erro tende a diminuir mais rapidamente, diminuindo com isto o tempo total de treinamento. Recomenda-se então o contrário, começar com um número grande de neurônios internos e ir reduzindo até obter o número adequado.

Em [45], são apresentadas algumas orientações sobre o número total de neurônios que devem compor a rede quando dispõe-se de um conjunto de treinamento bem definido. Este não é *a priori* o caso de algumas aplicações em controle, no entanto, estas regras podem servir como orientação. Segundo [45], uma rede multicamadas com função de ativação diferenciável é "global" e "absolutamente treinável" somente se

$$N \geq P + m - \text{rank}(X) \qquad (3.3)$$

onde N ---> número total de neurônios da rede
P ---> número de padrões de treinamento
m ---> número de saídas
X ---> conjunto dos padrões de entrada

ou

$$N_h \geq P - \text{rank}(X)$$

onde N_h é o número de neurônios internos.

De acordo com a equação (3.3), o número de neurônios necessários para que a rede possa ser realmente treinada deve aumentar principalmente com o aumento do número de neurônios de saída e do número de padrões a serem assimilados no treinamento, e deve aumentar quanto menor for o número de padrões de treinamento linearmente independentes.

A mesma referência prevê que se uma rede com N_g neurônios é "global" e "absolutamente treinável", então, a mesma rede com um número maior que N_g também será "global" e "absolutamente treinável".

Podemos utilizar estes conceitos para determinar o número total de neurônios da rede, e a partir disto, determinar a disposição dos mesmos em termos de camadas. Recomenda-se utilizar camadas internas de igual tamanho para obter uma melhor distribuição na hora da propagação dos sinais e dos erros.

3.5.2.3 - Parâmetros de treinamento

Quando utilizamos o algoritmo de treinamento *back-propagation*, os parâmetros de treinamento a serem definidos são: a taxa de treinamento η, a taxa de momento α, o erro total mínimo e o número total de iterações.

Estes podem ser definidos corretamente após uma ou várias tentativas iniciais de treinamento, visto que dependem do comportamento do algoritmo, isto é, da sua velocidade e convergência.

Após uma primeira simulação, observando os valores apresentados pelo erro total ao longo do treinamento, e comparando seu valor com o resultado apresentado na saída, podemos primeiro, definir um valor de erro tal que as saídas da rede apresentem uma resposta aceitável. Em seguida, podemos definir, de acordo com a velocidade de diminuição do valor do erro, um número de iterações aproximado no qual se espera que o erro chegue perto do valor mínimo desejado. O fim do treinamento, então, se dará quando o erro atingir seu valor mínimo ou quando chegarmos ao número de iterações definido (a primeira delas). Observando novas seqüências de treinamento, podemos definir valores mais adequados visando a diminuição do erro mínimo e do número de iterações necessárias para um bom resultado.

Já para as duas taxas (η e α), devemos lembrar que quanto maior for o seu valor, maiores serão as mudanças nos pesos, no entanto corre-se o risco de criar oscilações. Valores muito pequenos, não transmitem conhecimento nemhum à rede. Geralmente, escolhe-se valores altos para as primeiras tentativas, e diminue-se gradualmente até que sejam eliminadas as oscilações.

A faixa válida para as taxas está situada entre 0,1 e 1,0. Valores acima de 1 não fazem muito sentido, visto que poderão provocar mudanças exageradas provocando instabilidade.

3.5.2.4 - Conjunto de padrões para treinamento

No final do item 3.2, foram realizadas algumas considerações sobre como definir os sinais de treinamento para aplicações em controle.

A escolha ou definição do conjunto deverá levar em conta três características: consistência, representatividade e caracterização. A consistência refere-se à coerência dos padrões com o comportamento normal do sistema. A representatividade implica em que aquele conjunto de treinamento leve, efetivamente, o erro para o seu valor mínimo. E, finalmente, a caracterização refere-se à distribuição uniforme das várias entradas e saídas dentro do domínio de atuação do sistema.

Deve-se lembrar que quanto melhor definidos os padrões melhor será a resposta da rede para situações não incluídas no conjunto.

3.5.3 - Implementação

Definidos os parâmetros de treinamento, a seqüência a ser seguida é a do algoritmo de treinamento mostrada no item 2.4.5. Para um procedimento de identificação ou controle a seqüência está nos itens 3.2 e 3.3.1, respectivamente.

Os problemas mais comuns que podem surgir durante o treinamento são: ocorrência de mínimos locais e ausência de treinamento.

A primeira solução que surge na ocorrência de um destes problemas é a de recomeçar o treinamento, o que significa simplesmente em inicializar novamente os pesos com novos valores aleatórios. Mas, antes disso, deve-se tentar detectar a causa do problema para iniciar o novo treinamento sob outras condições. Do ponto de vista geométrico, esta reinicialização significa que a rede muda seu ponto de partida.

Esta parace uma solução muito fácil, no entanto, no caso dos mínimos locais, perde-se com ela todo o conhecimento adquirido até esse ponto. Estamos utilizando então um método de tentativa e erro, o que não é recomendado. Deve-se, portanto, descobrir o problema e tentar corrigí-lo.

A etapa de implementação consiste no treinamento de várias redes sob várias condições diferentes de ajuste dos parâmetros procurando o melhor resultado.

Uma solução menos radical, na ocorrência de mínimos locais é sugerida em [46], a qual consiste em dar um pequeno "empurrão" para tirar a rede desse ponto. Dar um "empurrão" consiste em modificar ligeiramente todos os pesos, mudando levemente o comportamento da mesma. Restaria definir com que intensidade e em que direção seria essa pequena mudança.

Após o treinamento é necessário realizar testes para verificar o desempenho do sistema. Para isto, devem-se testar tanto os padrões utilizados no treinamento como padrões diferentes, padrões novos, para verificar a capacidade de generalização da rede.

De acordo com o resultado dos testes, caso não sejam totalmente satisfatórios, pode-se proceder a um novo treinamento, levando-se em conta os novos requisitos ou as deficiências detectadas na hora de definir novos parâmetros de treinamento.

3.5.4 - Manutenção

A manutenção envolve modificações na rede devido à queda de precisão ou outros problemas detectados através de uma avaliação periódica ou através da monitoração contínua do desempenho.

Aconselha-se manter várias cópias de uma rede para futuras manutenções. Uma primeira cópia é mantida operando o sistema. A segunda pode ser utilizada para propósitos de desenvolvimento e

manutenção, e outras cópias podem ser utilizadas para novas análises e/ou pesquisas ou simplesmente como rede de reserva.

CAPÍTULO 4

SIMULAÇÃO DO CONTROLE DE VELOCIDADE UTILIZANDO REDES NEURONAIS

4.0 - INTRODUÇÃO

Segundo a metodologia de desenvolvimento estabelecida no item 3.5 do capítulo anterior, o primeiro passo consiste em descrever o problema a ser resolvido. O objetivo da aplicação escolhida neste estudo já foi apresentada no item 3.5.1. Resta apresentar alguns critérios que confirmem a utilização de redes neuronais para atingir tais objetivos.

O processo em questão é o acionamento de um motor de corrente contínua. Ele pode apresentar um comportamento altamente não-linear, o que dificulta a modelagem correta na hora de projetar um controlador. E, em certos casos, poderá haver uma forte influência de interferências produzidas no ambiente de trabalho.

Os sinais mais importantes do processo podem ser fácilmente coletados através de sensores, podendo estes estar contaminados por ruído. Podemos esperar, portanto, que a utilização de redes neuronais, formando uma estrutura de controle de tipo adaptativa, permita uma automatização mais fácil do procedimento de ajuste e de controle da malha de velocidade do motor, e aumente a imunidade do sistema ao ruído.

A solução do problema será dividida em quatro partes. Na primeira, o modelo do acionamento do motor CC é preparado para a realização das simulações digitais. Na partes seguintes serão especificadas e simuladas as estruturas de cada um dos três

esquemas que vão participar do sistema de controle: identificação, controle e monitoração (figuras 3.2, 3.6 e 3.4 respectivamente). No item 4.4, o esquema de monitoração será especificado e testado para verificar as caracteristicas da resposta e a capacidade de adaptabilidade.

Equipamento disponível

Para realizar todo o conjunto de simulações será utilizado um computador PC AT 386 de 33MHz.

Software disponível

Para as simulações e geração de resultados e gráficos será utilizado o pacote NEUWORK de redes neuronais, desenvolvido pelos autores e colaboradores deste trabalho, o qual é constituído por três programas:

- O NEUWORK I para criação de arquivos de redes, treinamento de tipo *off-line* através da criação de padrões de treinamento fixos e geração de resultados;

- O NEUWORK II utilizado para aplicações de tipo *on-line*, implementando as funções de identificação de processos e treinamento de controladores neuronais. O programa permite o acompanhamento do treinamento monitorando os sinais mais importantes através de valores ou em termos de gráfico, assim como o acompanhamento das mudanças nos valores dos pesos para análise de comportamento a nível interno;

- E o NEUCONTROL que é um aplicativo direcionado à análise de controladores em uma malha de velocidade, permitindo inclusive a utilização de um controlador neuronal treinado com o NEUWORK II. O programa inclui rotinas de teste de variação de parâmetros e presença de ruído na realimentação para fins de estudo da

resposta.

O apêndice B apresenta algumas das telas dos programas citados.

4.1 - SIMULAÇÃO DO ACIONAMENTO DO MOTOR CC

Com o surgimento, em grande escala, dos microcomputadores, tornou-se muito mais fácil o projeto e a análise dos sistemas de controle realimentados, não apenas na realização dos cálculos e simulações do sistema, mas também, na própria implementação.

As simulações permitem estudos detalhados, e, principalmente, confiáveis, levando a uma diminuição dos custos e do tempo de desenvolvimento.

Em geral, a simulação digital, como é conhecida, não ocorre em tempo real, no entanto, a validade dos resultados obtidos tem sido mais que comprovada.

Um processo analógico pode ser simulado em computador, uma vez que sua dinâmica pode ser aproximada por uma função de transferência no domínio Z. A análise consiste usualmente de dois passos :

1.- Representação do sistema contínuo por um modelo digital;

2.- Simulação do modelo digital em um computador digital.

Para representar um modelo na forma digital é inserido um dispositivo de amostragem e retenção ("*sample-hold*") no sistema contínuo para depois ser executada a discretização. Este procedimento será explicado de forma breve a seguir.

4.1.1 - Transformada Z

A transformada Z constitui uma das ferramentas matemáticas mais importantes para o estudo de sistemas digitais. O papel da transformada Z para os sistemas digitais é semelhante ao da transformada de Laplace para os sistemas contínuos.

Para tomar a transformada Z de um sinal contínuo f(t), seguem-se três passos :

1.- f(t) é amostrado para obter f*(t);

2.- Toma-se a transformada de Laplace de f*(t)

$$F^*(s) = \mathcal{L}\,[f^*(t)] = \sum_{k=0}^{\infty} f(kT)\cdot e^{-kTs}$$

onde T é o período de amostragem;

3.- Substitui-se e^{Ts} por z em F*(s) para obter F(z)

$$F(z) = \sum_{k=0}^{\infty} f(kT).z^{-k}$$

O Apêndice A apresenta uma tabela com a transformada Z de algumas funções comumente utilizadas.

A transformada Z de um sistema pode ser obtida facilmente a partir da função de transferência em Laplace do mesmo. Para isto, a função em *s* é expandida, quando possível, em frações parciais para então obter a transformada Z de cada uma das frações em separado, para depois somá-las. Isto é válido de acordo com a propriedade da linearidade da transformada Z mostrada também no Apêndice A.

O exemplo 1, apresentado a seguir, mostra como obter a transformada Z de uma função no domínio *s*.

Exemplo 1 : Obter a transformada Z da função :

$$E(s) = \frac{5}{s(s+1)(s+5)}$$

Solução : O primeiro passo consiste em separar a função E(s) em várias funções que possam ser encaixadas dentro das transformadas do Apêndice A.

$$E(s) = \frac{1}{s} - \frac{1,25}{s+1} + \frac{0,25}{s+5}$$

Aplicando a propriedade da linearidade,

$$E(z) = \frac{z}{z-1} - \frac{1,25z}{z-e^{-T}} + \frac{0,25z}{z-e^{-5T}}$$

Podemos observar que a função E(z) depende do período de amostragem T.

Mais detalhes ou mais exemplos sobre a transformada Z podem ser obtidos na literatura especializada [47, 48, 49].

4.1.2 - Equação as diferenças

Os sistemas de controle amostrados podem ser modelados por equações as diferenças. Para modelar digitalmente o comportamento de um sistema contínuo ou amostrado é necessário discretizar as variáveis contínuas.

A equação as diferenças que representa o comportamento de um sistema pode ser obtida a partir da função de transferência em Z do sistema amostrado, como mostra o exemplo 2.

Exemplo 2 : Dada a função de transferência em Z do sistema amostrado

$$\frac{C(z)}{R(z)} = \frac{2z^2 + 0,4z}{z^2 - 0,4z + 0,2}$$

obter a equação as diferenças.

Solução : Para obter a equação as diferenças devemos primeiro manipular a função dada:

$$\frac{C(z)}{R(z)} = \frac{2 + 0,4z^{-1}}{1 - 0,4z^{-1} + 0,2z^{-2}}$$

Manipulando novamente,

$$(1 - 0,4z^{-1} + 0,2z^{-2}) \cdot C(z) = (2 + 0,4z^{-1}) \cdot R(z)$$

Utilizando a propriedade do deslocamento da transformada Z, e tomando a transformada inversa da equação anterior teremos

$$c(kT) - 0,4c[(k-1)T] + 0,2c[(k-2)T] = 2r(kT) + 0,4r[(k-1)T]$$

$$c(kT) = 2r(kT) + 0,4r[(k-1)T] + 0,4c[(k-1)T] - 0,2c[(k-2)T]$$

ou simplesmente

$$c(k) = 2r(k) + 0,4r(k-1) + 0,4c(k-1) - 0,2c(k-2)$$

onde *r* é o sinal de excitação e *c* é a saída do sistema.

4.1.3 - Zero-order hold (ZOH)

O *zero-order hold* é um dispositivo de retenção, cuja finalidade é a de converter um simples impulso de entrada em uma onda aproximadamente retangular. Pretende-se com isto reconstruir um sinal amostrado no sinal contínuo correspondente. Um circuito capaz de realizar esta tarefa é apresentado na figura 4.1, que, como pode ser visto, é sempre associado a um dispositivo de amostragem.

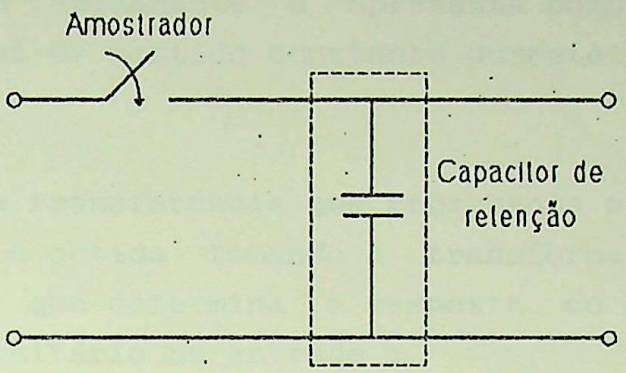
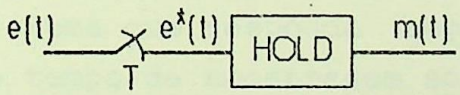


Figura 4.1 - Amostragem + Dispositivo de retenção.

Na figura 4.2, vemos, através de gráficos, um exemplo de conversão de um sinal constituído de impulsos em um sinal contínuo através do *zero-order hold*. Sendo que e*(t) representa valores amostrados de e(t), podemos dizer que se trata de um processo de reconstrução.



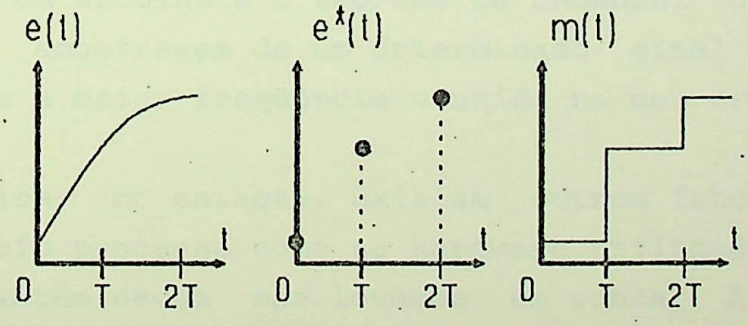


Figura 4.2 - Funcionamento do "Sample-Hold".

Outro exemplo de reconstrução, precisamente aquele que mais nos interessa, é a conversão do sinal de saída discreto de um computador digital em um sinal contínuo. Quando analisado dentro deste contexto, o ZOH é considerado parte do computador. Isto deve-se ao fato de que o computador amostra vários sinais,

executa cálculos rapidamente e apresenta como saída o sinal manipulado, o qual é mantido constante durante todo o intervalo de amostragem.

A função de transferência que representa o funcionamento do *zero-order hold* é obtida tomando a transformada de Laplace da seguinte função, que determina a resposta do dispositivo a um simples impulso unitário na entrada :

$$g_{ZOH}(t) = U_{-1}(t) - U_{-1}(t-T)$$

$$G_{ZOH}(s) = \frac{1 - e^{-Ts}}{s}$$

onde T é o período de amostragem.

4.1.4 - Escolha do período de amostragem

O período de amostragem deverá ser o menor possível, afim de se obter um sistema que responda rapidamente e minimize os efeitos adversos do tempo de amostragem sobre a estabilidade.

A base da escolha é o teorema de Shannon. Segundo este, a freqüência de amostragem de um determinado sinal deve ser maior que duas vezes a maior freqüência contida no seu espectro.

Na prática, no entanto, existem outros fatores associados tanto ao próprio processo como ao hardware utilizado na implementação, que também devem ser levados em conta. As técnicas de projeto de controladores digitais, principalmente as de resposta rápida, "deadbeat controllers", dependem diretamente do período de amostragem.

O tempo total de computação, isto é, os tempos de conversão A/D e D/A e o tempo envolvido nas operações aritméticas e outras funções, deve ser determinado, afim de que a soma destes seja razoavelmente menor do que o período de amostragem a ser adotado.

Uma análise mais profunda envolveria ainda outros parâmetros também importantes, entre eles, os possíveis atrasos contidos no sistema e as constantes de tempo presentes nos transdutores. O comprimento da palavra digital utilizada, isto é, o número de bits utilizado nos circuitos digitais, tem também estreita relação com a escolha.

Emfim, levando em conta todos os fatores acima relacionados, podemos chegar a especificar uma freqüência de amostragem de até 8 ou 10 vezes a freqüência mínima especificada pelo teorema de Shannon.

As simulações constituem hoje uma ferramenta muito poderosa na iniciativa de diminuir cada vez mais o período de amostragem, tentando aproximar na medida do possível os sistemas digitais dos sistemas contínuos. Deve-se notar entretanto, que a freqüência de funcionamento dos circuitos utilizados em sistemas de controle, principalmente os microprocessadores, está associada diretamente ao custo dos mesmos, ou seja, quanto maior a freqüência de trabalho permitida maior o custo. Logo, um estudo adequado poderá significar em economia de custos.

Para o caso específico de um sistema de controle de velocidade de um motor CC a escolha deve ser feita levando em consideração, basicamente, a constante de tempo mecânica do motor, que representa o fator mais crítico. Comumente, utiliza-se um período de amostragem dez vezes menor que a constante de tempo mecânica.

4.1.5 - Equação as diferenças do acionamento

O diagrama de blocos utilizado como base para a simulação do processo é mostrado na figura 4.3, onde o ZOH é colocado para representar o manuseio digital do processo através de um conversor A/D.

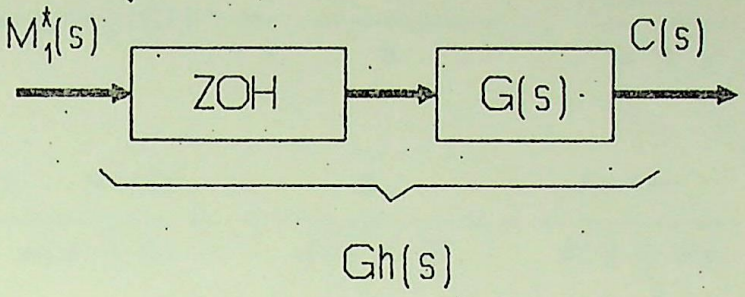


Figura 4.3 - Diagrama de blocos para simulação do processo.

O efeito do ZOH é agrupado e processado como parte do processo G(s) na função $G_h(s)$.

A função G(s), que representa a função de transferência do processo, foi definida no item 1.2.4, na equação (1.24):

$$G(s) = \frac{Vt}{Vc}(s) = \frac{Kc\left(\frac{Ka\phi}{Ra \cdot B}\right)Kt}{s\tau_m + \left(1 + \frac{ka\phi}{Ra \cdot B}\right)}$$

A função anterior pode ser expressa, para facilitar os cálculos, da seguinte forma:

$$G(s) = \frac{Vt}{Vc}(s) = \frac{A}{s + D} \qquad (4.1)$$

onde $A = \frac{Kc \cdot Ka\phi \cdot Kt}{\tau_m \cdot Ra \cdot B}$ e $D = \frac{1}{\tau_m}\left(\frac{Ra \cdot B + Ka\phi^2}{Ra \cdot B}\right)$

Desta forma, a função $G_h(s)$ fica sendo:

$$G_h(s) = G_{ZOH}(s) \cdot G(s) = \frac{1 - e^{-Ts}}{s} \cdot \frac{A}{s + D} \qquad (4.2)$$

$$G_h(s) = \frac{A - A \cdot e^{-Ts}}{s(s + D)} = \frac{A}{s(s + D)} - \frac{Ae^{-Ts}}{s(s + D)} \qquad (4.3)$$

A transformada da função $G_h(s)$ é obtida da mesma forma como foi mostrado no exemplo 1 deste item.

A equação (4.3) pode ser expressa de forma mais simples:

$$G_h(z) = Z[G_1(s)] - Z[e^{-Ts} \cdot G_1(s)]$$

onde $G_1(s) = \frac{A}{s(s + D)}$

Utilizando a tabela de transformadas do Apêndice A, teremos:

$$G_h(z) = Z[G_1(s)] - z^{-1} \cdot Z[G_1(s)] \qquad (4.4)$$

Para obter G_h Devemos primeiro obter a transformada de $G_1(s)$:

$$Z[G_1(s)] = Z\left[\frac{A}{s(s + D)}\right] = Z\left[\frac{ao}{s} + \frac{a1}{s + D}\right]$$

$$= Z\left[\frac{A/D}{s} - \frac{A/D}{S + D}\right]$$

$$= \frac{A}{D} Z\left[\frac{1}{s}\right] - \left[\frac{A}{D}\right] Z\left[\frac{1}{s + D}\right]$$

$$= \frac{A}{D}\left[\frac{z}{z - 1}\right] - \frac{A}{D}\left[\frac{z}{z - e^{-DT}}\right]$$

$$= \frac{A}{D}\left[\frac{z}{z-1} - \frac{z}{z-e^{-DT}}\right]$$

Levando este resultado na equação (4.3) :

$$G_h(z) = (1 - z^{-1}) \cdot Z[G_1(s)]$$

$$= \left[\frac{z-1}{z}\right] \frac{A}{D}\left[\frac{z}{z-1} - \frac{z}{z-e^{-DT}}\right]$$

$$= \frac{A}{D}\left[1 - \frac{z-1}{z-e^{-DT}}\right] = \frac{A}{D}\left[\frac{1-e^{-DT}}{z-e^{-DT}}\right]$$

multiplicando numerador e denominador por z^{-1}, tem-se :

$$G_h(z) = \frac{A}{D}\left[\frac{z^{-1}}{z^{-1}} \cdot \frac{1-e^{-DT}}{z-e^{-DT}}\right]$$

$$G_h(z) = \frac{A}{D}\left[\frac{(1-e^{-DT}) \cdot z^{-1}}{1-e^{-DT} \cdot z^{-1}}\right] \qquad (4.5)$$

E, sabemos que

$$G_h(z) = \frac{C(z)}{M(z)}$$

Então, finalmente teremos :

$$\frac{C(z)}{M(z)} = \left[\frac{(1-e^{-DT}) \cdot z^{-1}}{1-e^{-DT} \cdot z^{-1}}\right] \qquad (4.6)$$

Manipulando agora a equação (4.6) como foi feito no exemplo 2 do item 4.1.2, chegaremos na equação as diferenças desejada :

$$(1 - e^{-DT})\cdot C(z) = \frac{A}{D}\,[(1 - e^{-DT})\cdot z^{-1}]\cdot M(z)$$

$$c(k) - e^{-DT}\cdot c(k-1) = \frac{A}{D}\,(1 - e^{-DT})\cdot m(k-1)$$

$$c(k) = e^{-DT}\cdot c(k-1) + \frac{A}{D}(1 - e^{-DT})\cdot m(k-1) \qquad (4.7)$$

Analisando a equação (4.7), vemos que esta depende dos parâmetros do processo e do período de amostragem T, escolhido de acordo com os critérios definidos no item 4.1.4.

Sendo assim, para poder realizar a simulação é necessário que sejam definidos valores numéricos para estes parâmetros. Para tal fim, adota-se um acionamento com as seguintes especificações:

Motor CC

1,2 [KW] 110 [V] 1500 [rpm]

Ra = 1,16 [Ω] La = 60 [mH] τ_e = 0,052 [s]

J = 0,023 [kg.m] B = 0,018 [N.m.s/rad]

τ_m = 1,28 [s]

$k_{e\phi}$ = 0,66 [V.s/rad]

Tacômetro

K_t = 0,29 [V.s/rad]

Conversor Tiristorizado

$$Kc = \frac{2,34(110)}{\sqrt{3}\cdot 6} = 25$$

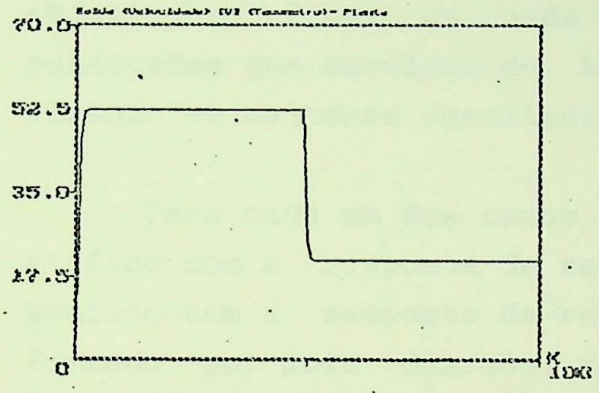
ou seja, uma tensão de alimentação de 110 [V] e uma tensão de controle máxima de 6 [V].

Levando estes valores na equação (4.7) e adotando um período de amostragem T de 200 milisegundos, teremos :

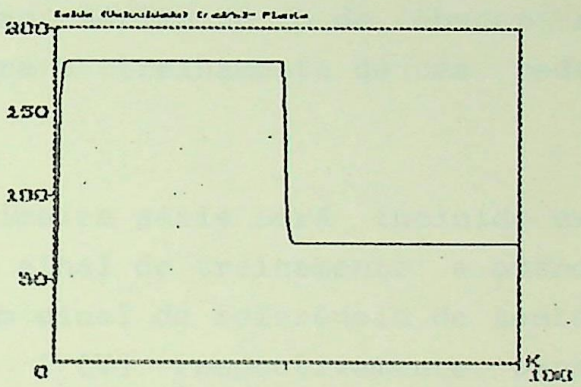
$$c(k) = 0,0328\cdot c(k-1) + 10,1379\cdot m(k-1) \qquad (4.8)$$

onde a excitação do sistema *m* representa a tensão de controle Vc que ativa o circuito de disparo, e *c* a saída do processo, ou seja, a saída do tacômetro.

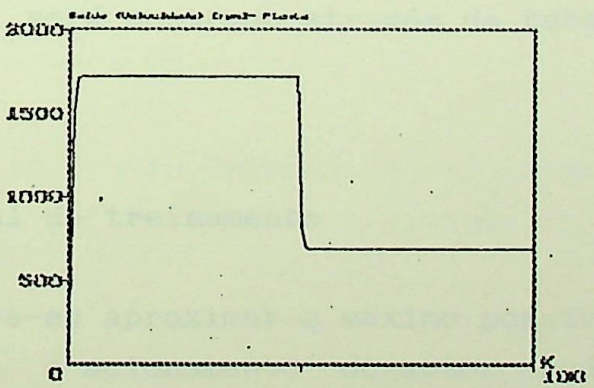
Na figura 4.4, apresenta-se a resposta do processo diante de uma sinal de excitação formado por dois degraus de 5[V] e 2[V], respectivamente. A figura 4.4(a) mostra a saída diretamente a partir da equação (4.8). A figura 4.4(b) mostra a saída equivalente antes do tacômetro, ou seja, a velocidade em [rad/s]. E, finalmente, a figura 4.4(c) mostra a mesma saída só que em [rpm].



(a)



(b)



(c)

Figura 4.4 - Simulação do processo.
(a) Saída do transdutor ([V]);
(b) Saída em [rad/s];
(c) Saída em [rpm].

4.2 - PROCEDIMENTO DE IDENTIFICAÇÃO - TESTES E RESULTADOS

Para simular o esquema da figura 3.2 devemos especificar a estrutura e parâmetros da rede e os parâmetros do treinamento, assim como definir também o sinal de treinamento a ser utilizado.

Em uma primeira série de treinamentos será fixado um número máximo de iterações para testar o grau de treinamento em termos do erro total final. E, em um segunda série, a partir dos resultados obtidos na primeira, será definido um erro total mínimo para análise da velocidade de queda do erro, verificando o número de iterações necessário para chegar a tal valor. O objetivo no final de cada uma das séries é o de chegar a conclusões que serviram de base para o treinamento de uma rede visando os melhores resultados.

Para cada um dos casos da primeira série será incluído um gráfico com a resposta da rede ao sinal de treinamento e outro gráfico com a resposta da rede a um sinal de referência de teste formado por dois degraus, de 5 e 2 [V] respectivamente, para facilitar a comparação entre as diversas redes. Em ambos gráficos será colocada a resposta do processo à mesma excitação para verificar os resultados em termos de identificação. Já para a segunda série de treinamentos basta analisar o número total de iterações, o qual será mostrado através de tabelas.

Definição do sinal de treinamento

O sinal deve-se aproximar o máximo possível da realidade. O sinal que excita o acionamento do motor pode ser formado por degraus ou rampas. O domínio de valores da entrada é o da tensão de controle Vc que alimenta o circuito de disparo, cujo valor máximo, dado pelo próprio circuito, foi definido no item 4.1.5 como sendo 6 [V].

Desta forma, o sinal poderá ser constituído por degraus e/ou rampas variando de forma pré-determinada ou aleatória na

faixa entre 0 e 6 [V].

A tensão de entrada máxima (6 [V]) corresponde a uma saída do transdutor de 62,9[V]. Sendo assim, podemos utilizar as faixas a seguir para a normalização dos valores operados pela rede: 0..6 para a entrada e amostras anteriores da referência, e 0..70 para a saída e realimentações da saída do processo. O tipo de normalização utilizado é aquele mostrado na equação (2.18).

O sinal de treinamento será especificado pelos dois parâmetros mostrados na figura 4.5, a duração de cada valor de referência *Dr* e o tempo de subida ou tempo de variação *Tv*. Na maioria dos casos será adotada uma duração *Dr*=30 iterações, o que equivale a 6 segundos com o período de amostragem de 200 ms.

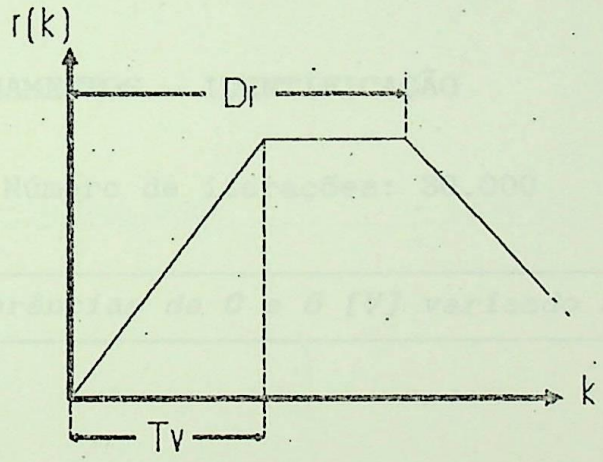


Figura 4.5 - Parâmetros de especificação do sinal de treinamento.

Definição da estrutura da rede

A orientação apresentada no capítulo 3 (item 3.5.2.2) sobre o tamhanho e estrutura da rede, especificamente a equação (3.3), é difícil de ser seguida neste caso, já que não existe um conjunto perfeitamente definido de padrões de treinamento. Os padrões são formados durante a fase de treinamento e dependem das

especificações da referência.

Segundo a equação (4.8) o acionamento do motor CC, com as simplificações realizadas na modelagem, corresponde a um sistema de 1ª ordem, logo podemos optar por utilizar o esquema da figura 3.2, inicialmente, com apenas uma amostra anterior de cada sinal. Outros treinamentos serão realizadas variando cada um dos parâmetros de treinamento e a estrutura da rede em termos de número de amostras anteriores e número de neurônios internos.

Mesmo que a rede, implementada em hardware, explore o paralelismo, um dos nossos objetivos é o de especificar a estrutura mais simples que atinja os resultados esperados, já que em sistemas mais complexos o tempo de processamento tem um papel importante.

1ª SÉRIE DE TREINAMENTOS - IDENTIFICAÇÃO

Número de iterações: 30.000

Sinal com referências de 0 e 6 [V] variando alternadamente.

Teste Nº1

Rede: 3 entradas, 1 camada interna (3 neurônios)

|

└——> 1 amostra anterior da referência e da saída

η : 0,5 α: 0,5

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.6; Erro de Identificação Final: 0,0169

Obs: - Este primeiro teste utiliza uma rede com estrutura mínima para servir como rede de referência.

- O cálculo do erro total é feito a cada 60 iterações, ou seja, a duração de duas variações na referência.

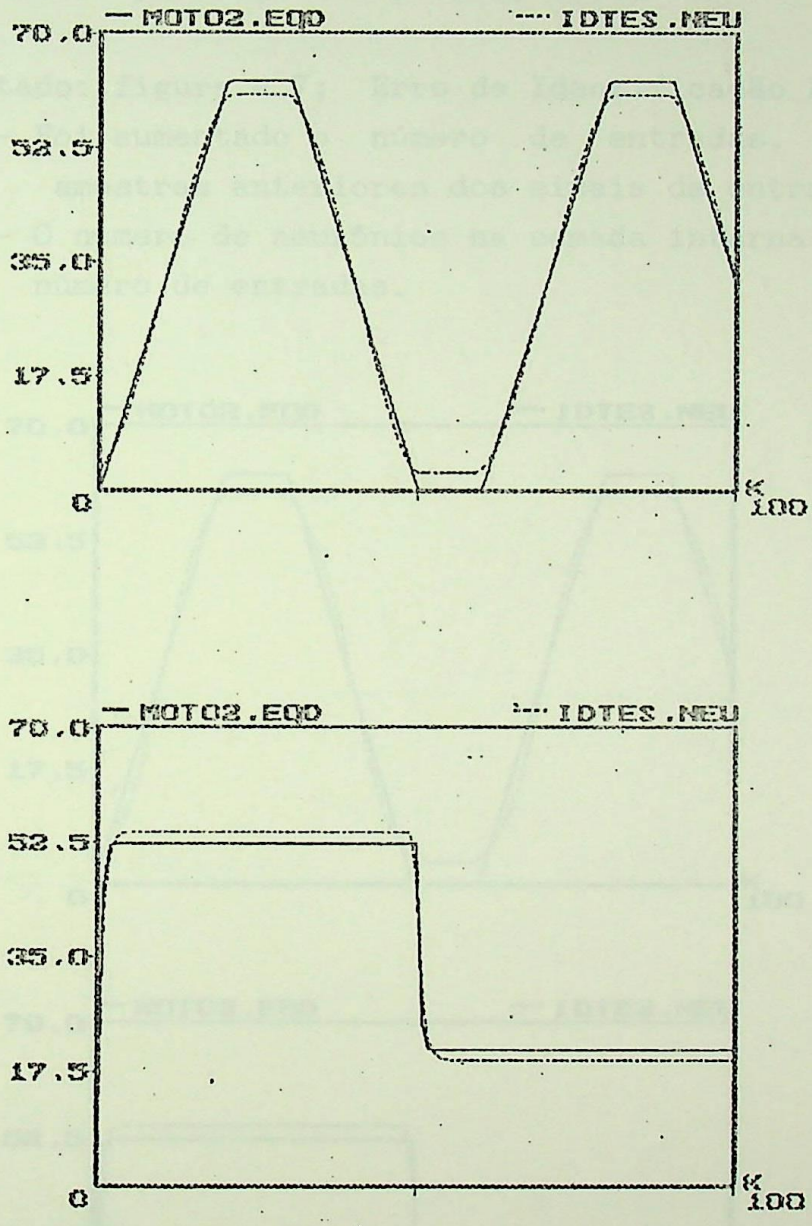


Figura 4.6 - Teste Nº1 - Identificação.

Teste Nº2

Rede: 5 entradas, 1 camada interna (5 neurônios)

└——> 2 amostras anteriores da referência e da saída

η : 0,5 α: 0,5

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.7; Erro de Identificação Final: 0,0451

Obs: - Foi aumentado o número de entradas, isto é, de amostras anteriores dos sinais de entrada.

- O número de neurônios na camada interna é igual ao número de entradas.

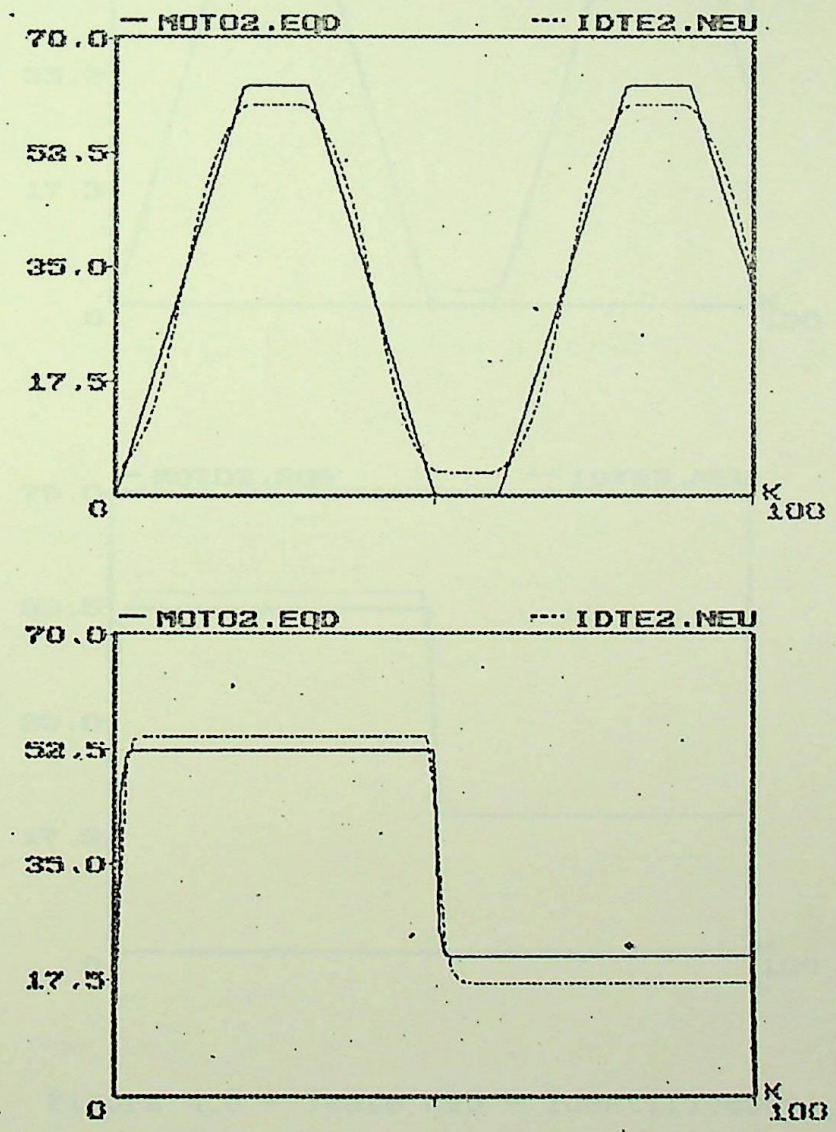


Figura 4.7 - Teste Nº2 - Identificação

Teste Nº3

Rede: 3 entradas, 2 camadas internas (3 neurônios cada)
η : 0,5 α: 0,5
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.8; Erro de Identificação Final: 0,0102
Obs: Foi aumentado o número de camadas internas.

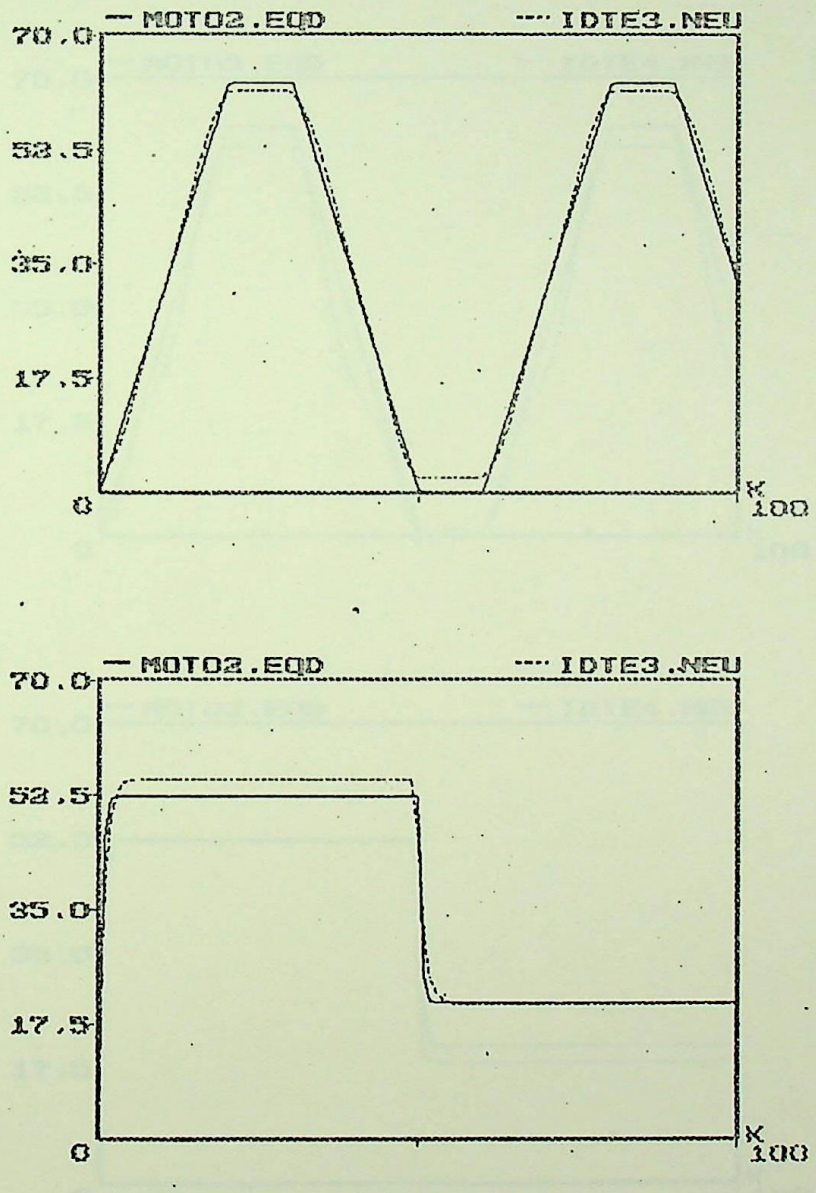


Figura 4.8 - Teste Nº3 - Identificação

Teste Nº4

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,8 α: 0,8

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.9; Erro de Identificação Final: 0,0356

Obs: Foi aumentado o valor das taxas de treinamento η e α.

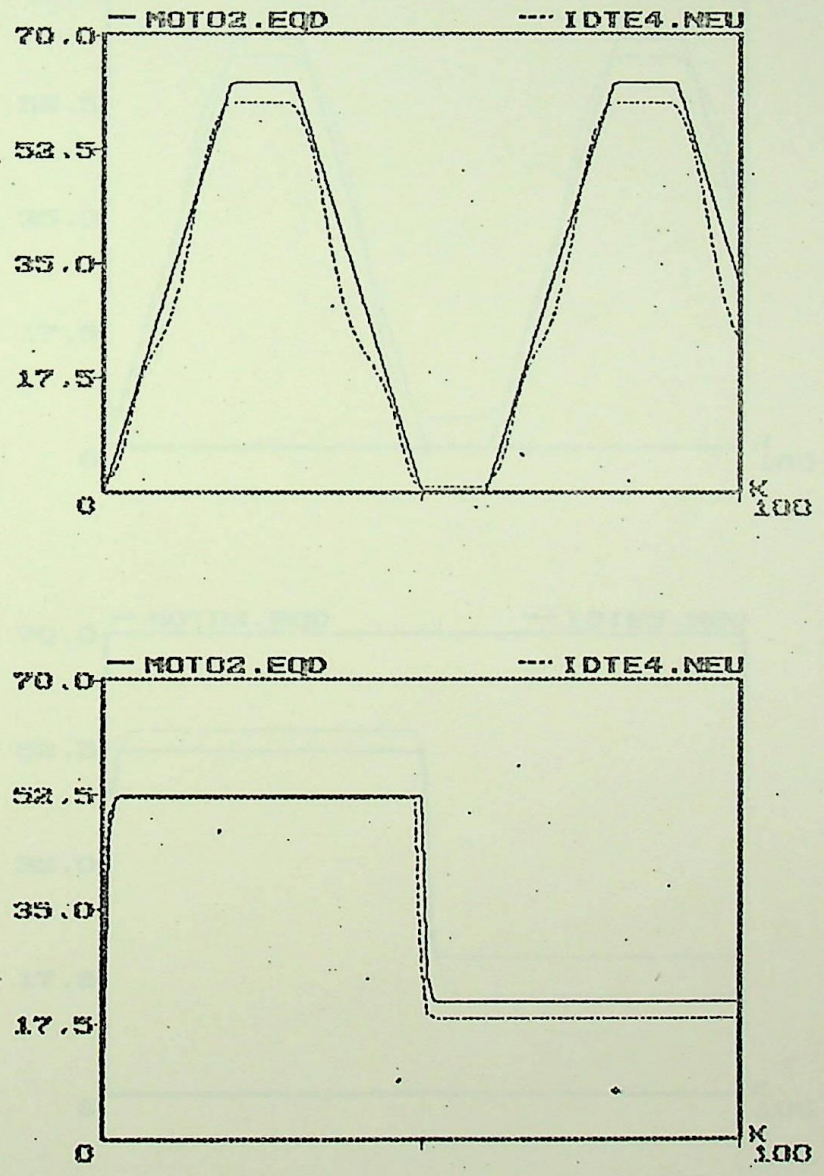


Figura 4.9 - Teste Nº4 - Identificação

Teste Nº5

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,3 α: 0,3

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.10; Erro de Identificação Final: 0,0490

Obs: Foi diminuido o valor das taxas de treinamento.

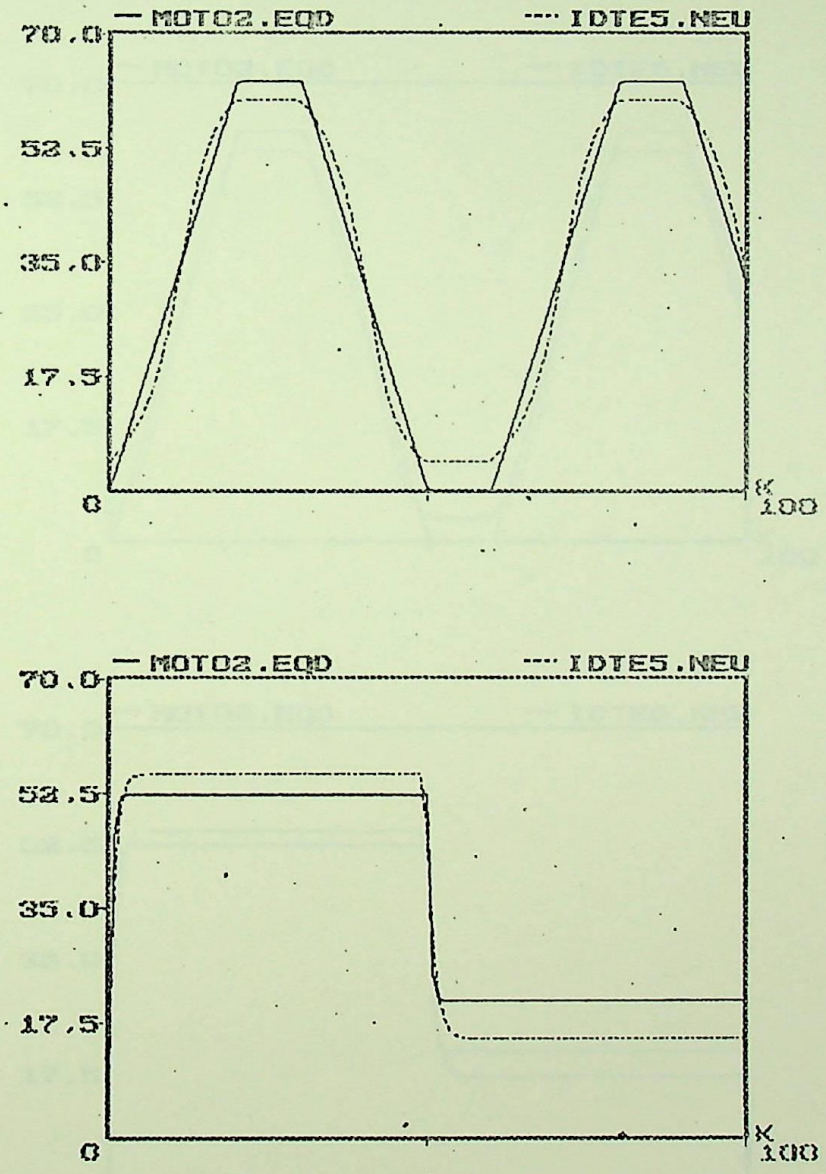


Figura 4.10 - Teste Nº5 - Identificação.

Teste Nº6

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,5 α: 0,2

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.11; Erro de Identificação Final: 0,0334

Obs: Verificação da ação da taxa de aprendizagem η .

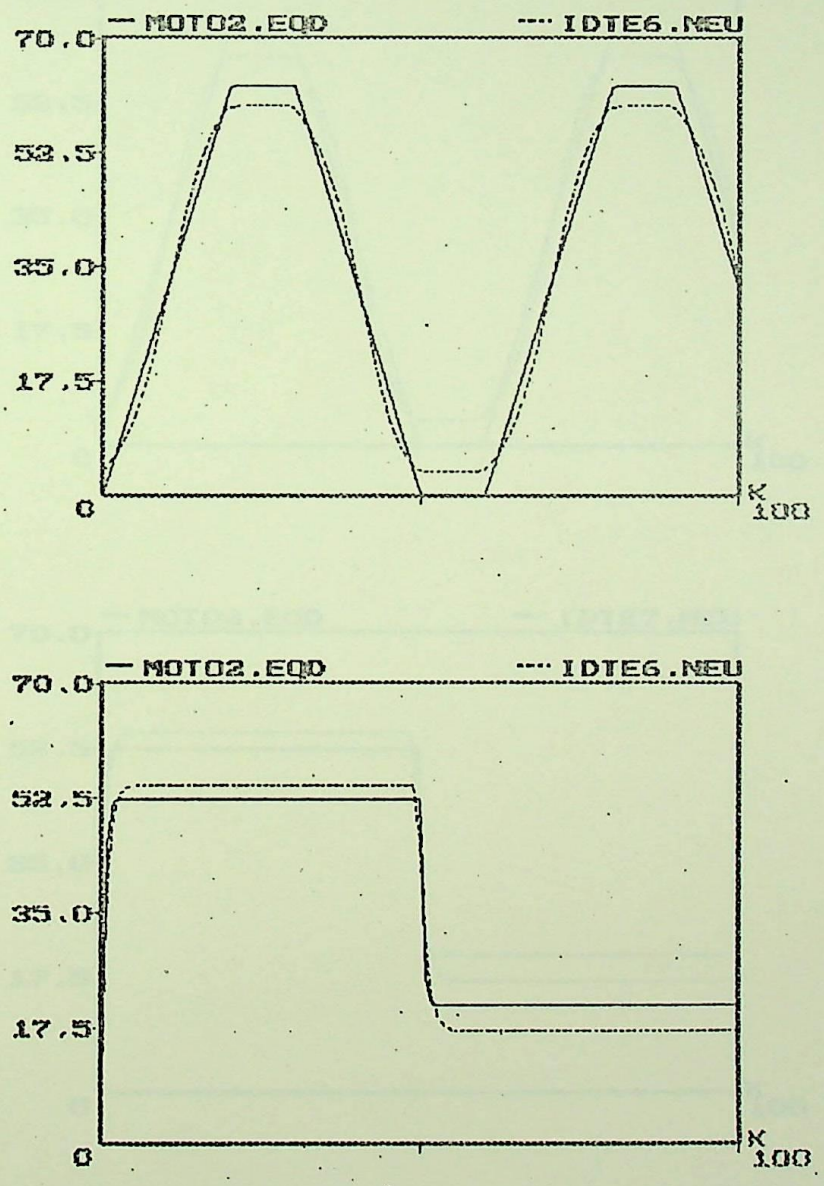


Figura 4.11 - Teste Nº6 - Identificação.

Teste Nº7

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,2 α: 0,5

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.12; Erro de Identificação Final: 0,0382

Obs: Verificação da ação da taxa de momento α.

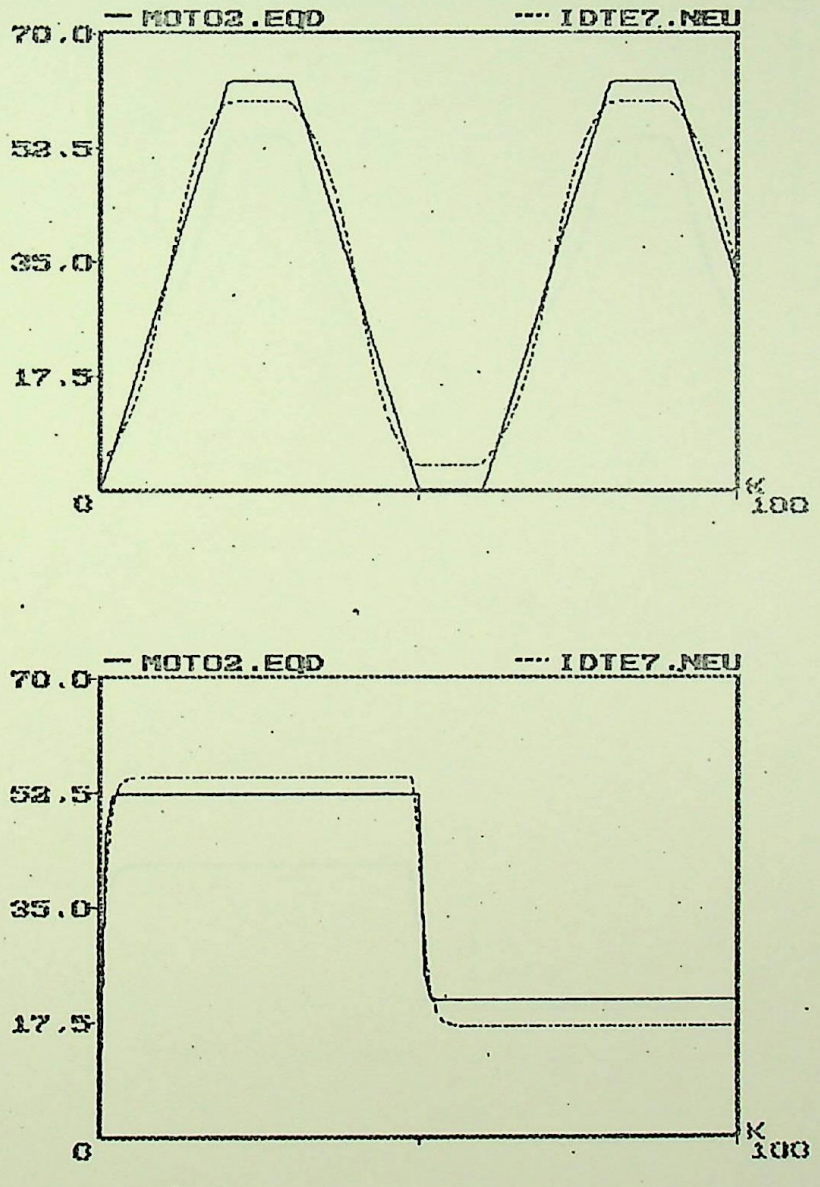


Figura 4.12 - Teste Nº7 - Identificação.

Teste Nº8

Rede: 3 entradas, 1 camada interna (3 neurônios)
η : 0,5 α: 0,8
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.13; Erro de Identificação Final: 0,0249
Obs: Foi aumentado o efeito da taxa de momento.

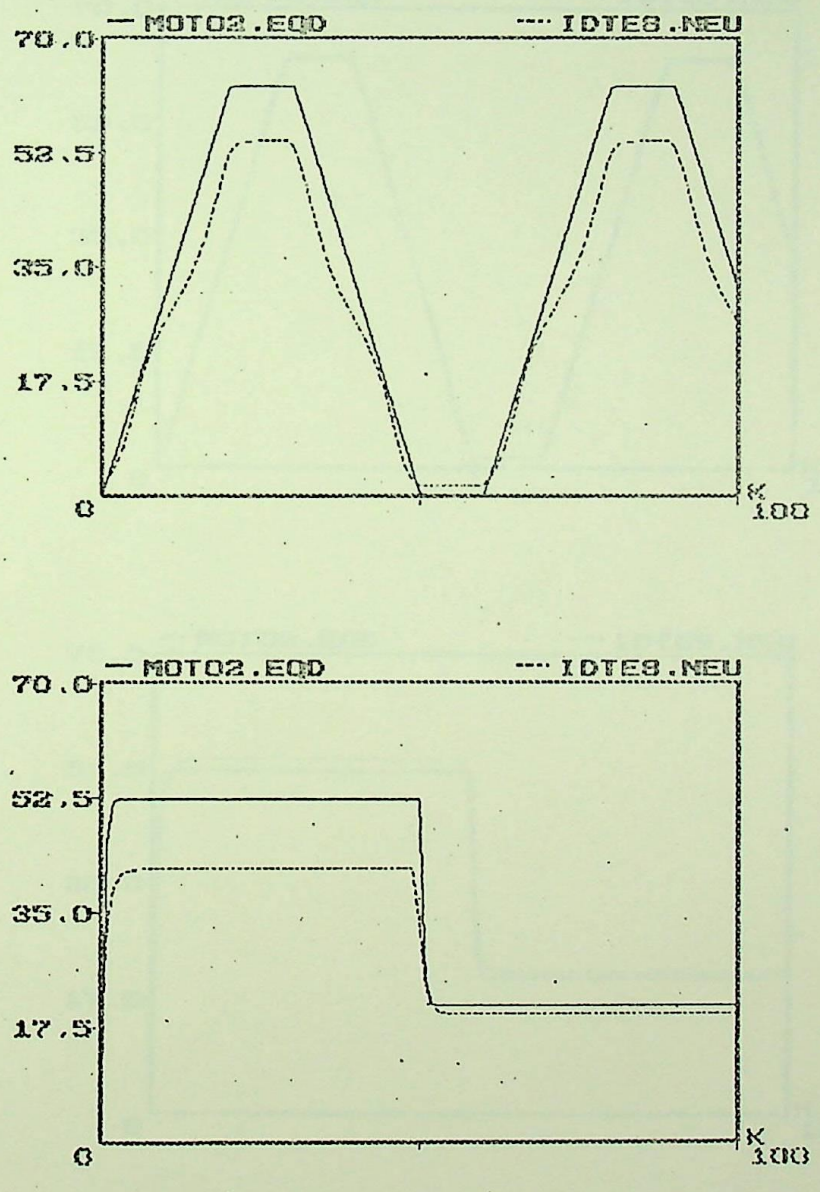


Figura 4.13 - Teste Nº8 - Identificação.

Teste Nº9

Rede: 3 entradas, 1 camada interna (3 neurônios)
η : 0,8 α: 0,5
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.14; Erro de Identificação Final: 0,0057
Obs: Foi aumentado o efeito da taxa de aprendizagem.

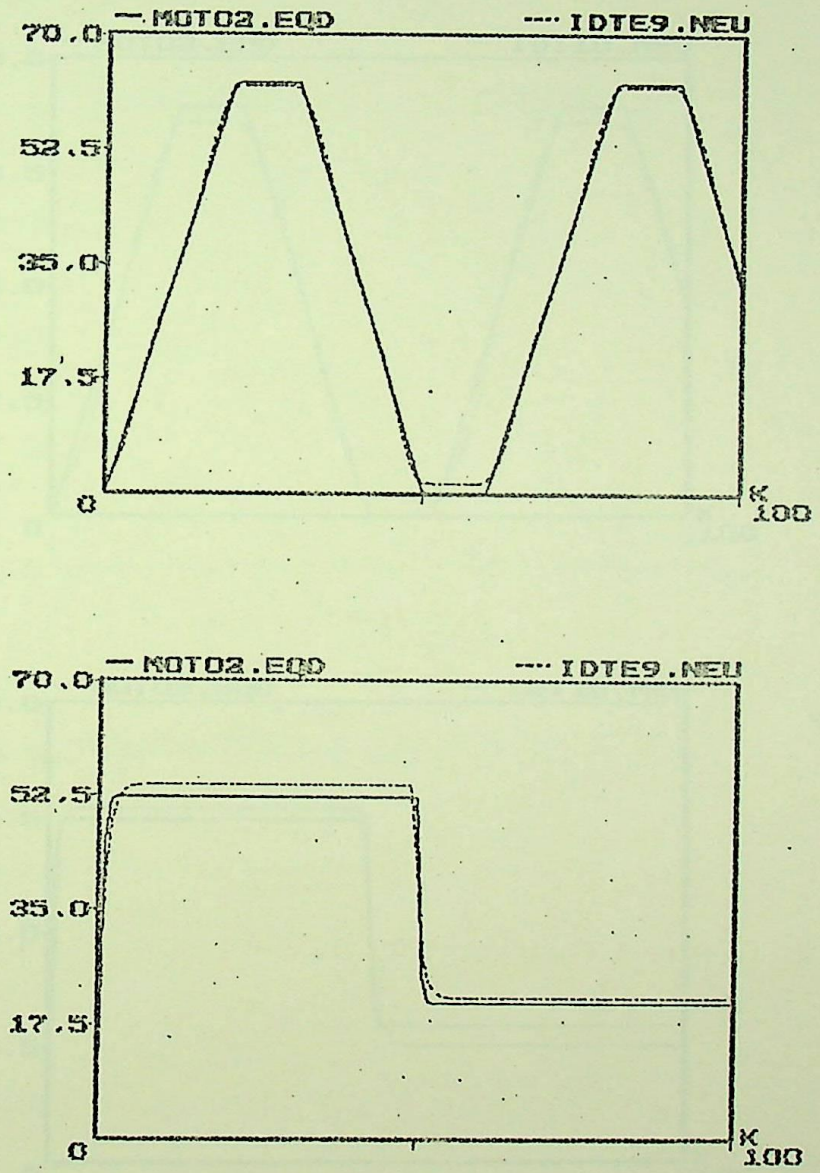


Figura 4.14 - Teste Nº9 - Identificação.

Teste Nº10

Rede: 5 entradas, 2 camadas internas (5 neurônios cada)
η : 0,5 α: 0,5
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.15; Erro de Identificação Final: 0,0284
Obs: Foi aumentado o número de camadas e neurônios internos.

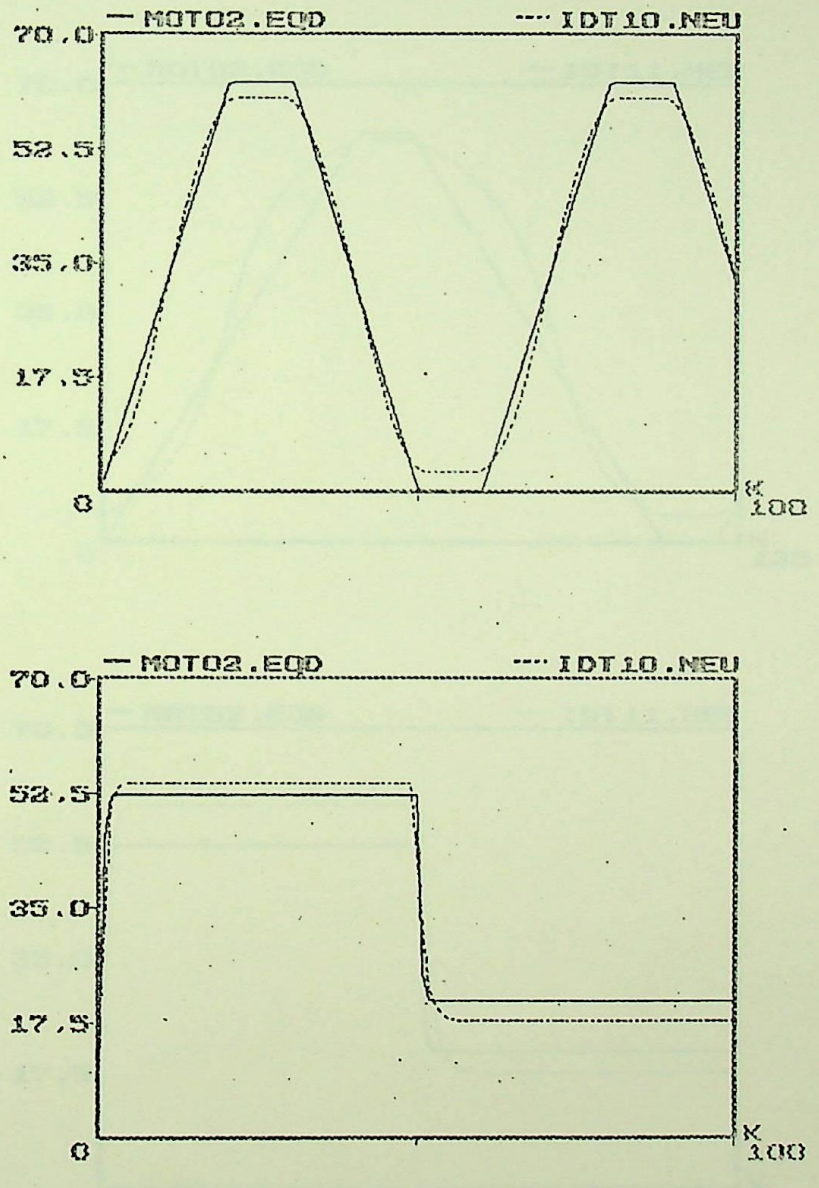


Figura 4.15 - Teste Nº10 - Identificação.

Teste Nº11

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,5 α: 0,5

Sinal: Dr=60 Tv=50

N. de iter. para cálculo do erro total: 120

Resultado: figura 4.16; Erro de Identificação Final: 0,0549

Obs: Foi aumentada a duração e tempo de variação da referência, tendo então menos perturbações.

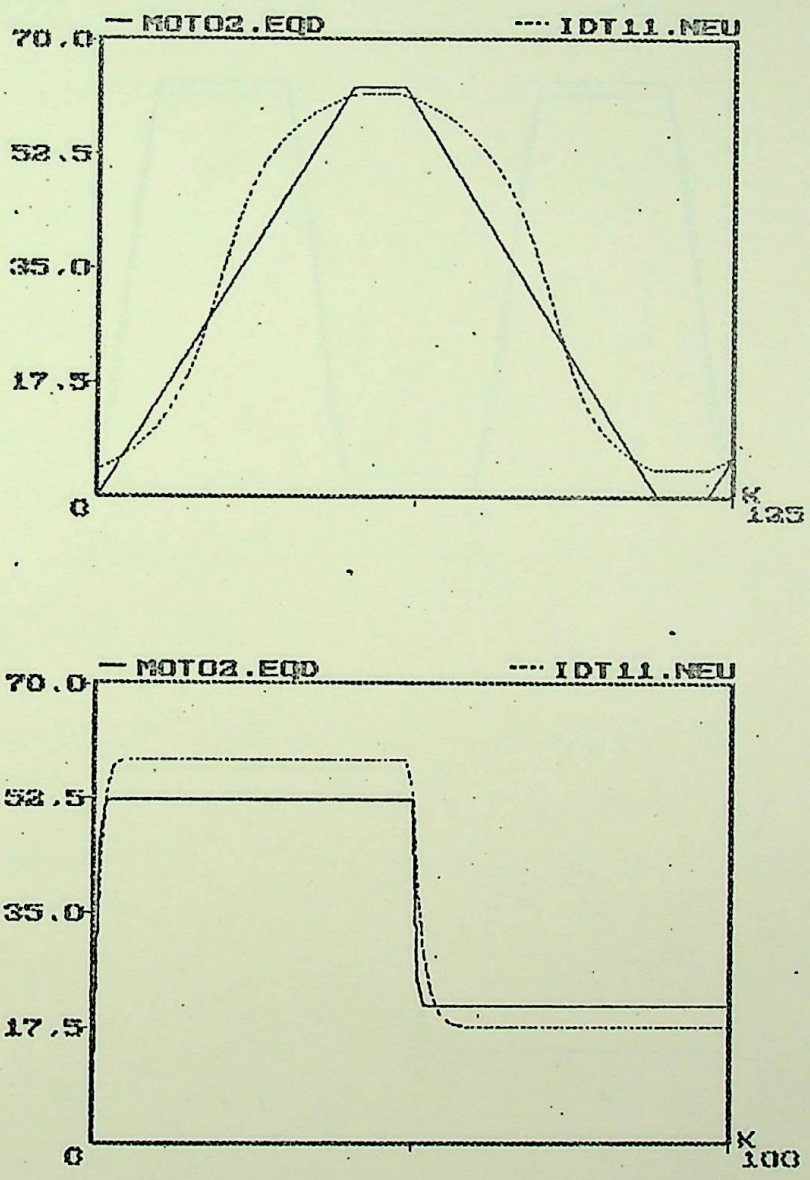


Figura 4.16 - Teste Nº11 - Identificação.

Teste Nº12

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,5 α: 0,5

Sinal: Dr=30 Tv=10

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.17; Erro de Identificação Final: 0,0411

Obs: Foi aumentada apenas a taxa de variação da referência para aproximá-la mais de um degrau.

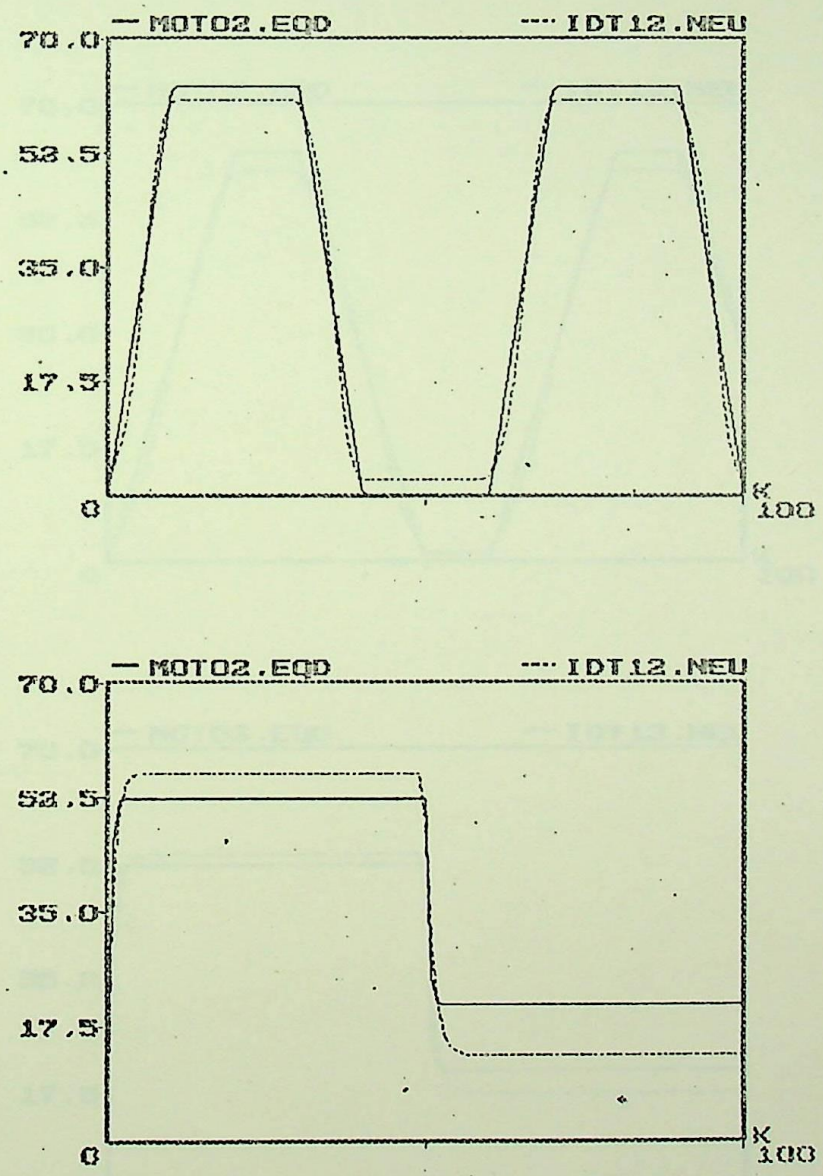


Figura 4.17 - Teste Nº12 - Identificação.

Teste Nº13

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,8 α: 0,5 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.18; Erro de Identificação Final: 0,0098

Obs: *Mudança na normalização*

Valores anteriores: 0..6 para a referência

0..70 para a saída

Novos valores: -1..7 para a referência

-10..80 para a saída

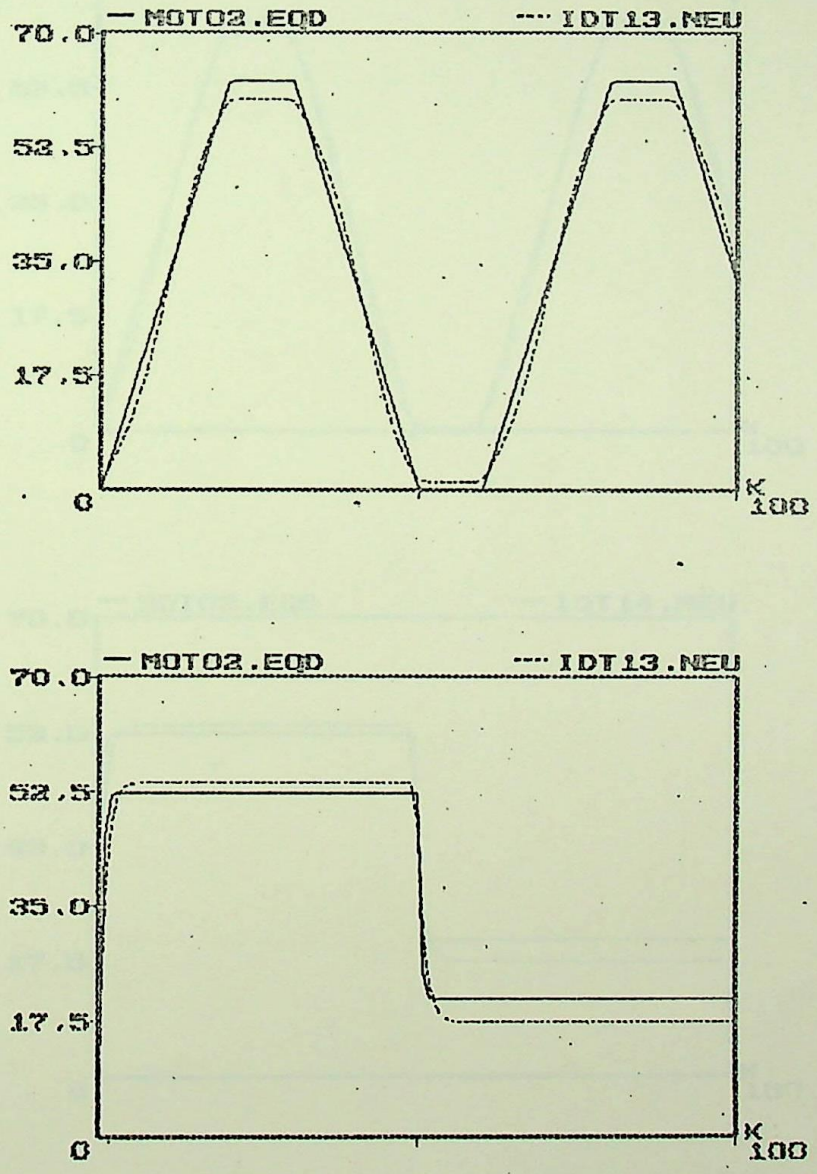


Figura 4.18 - Teste Nº13 - Identificação.

Teste Nº14

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,8 α: 0,5 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.19; Erro de Identificação Final: 0,0037

Obs: *Mudança na normalização*

Novos valores: -2..8 para a referência

-20..90 para a saída

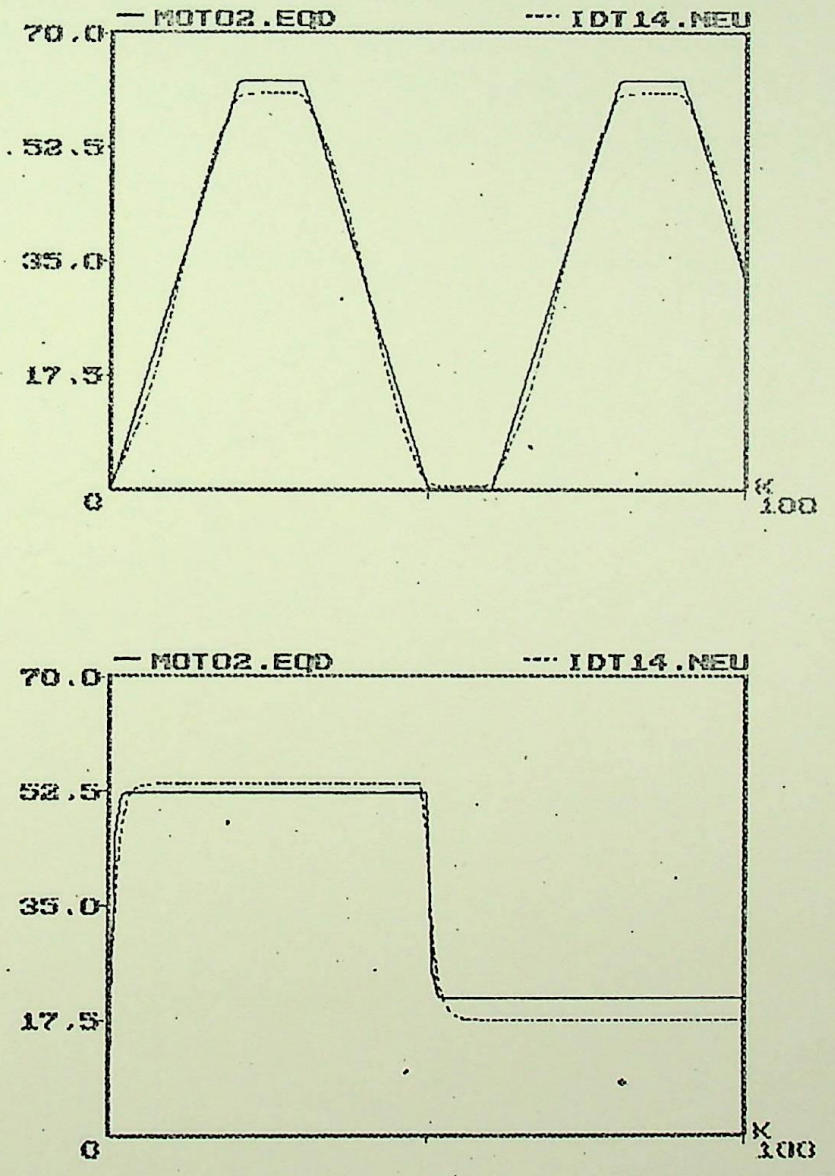


Figura 4.19 - Teste Nº14 - Identificação.

Teste Nº15

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,8 α: 0,5 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.20; Erro de Identificação Final: 0,0021

Obs: *Mudança na normalização*

Novos valores: -3..9 para a referência

-30..100 para a saída

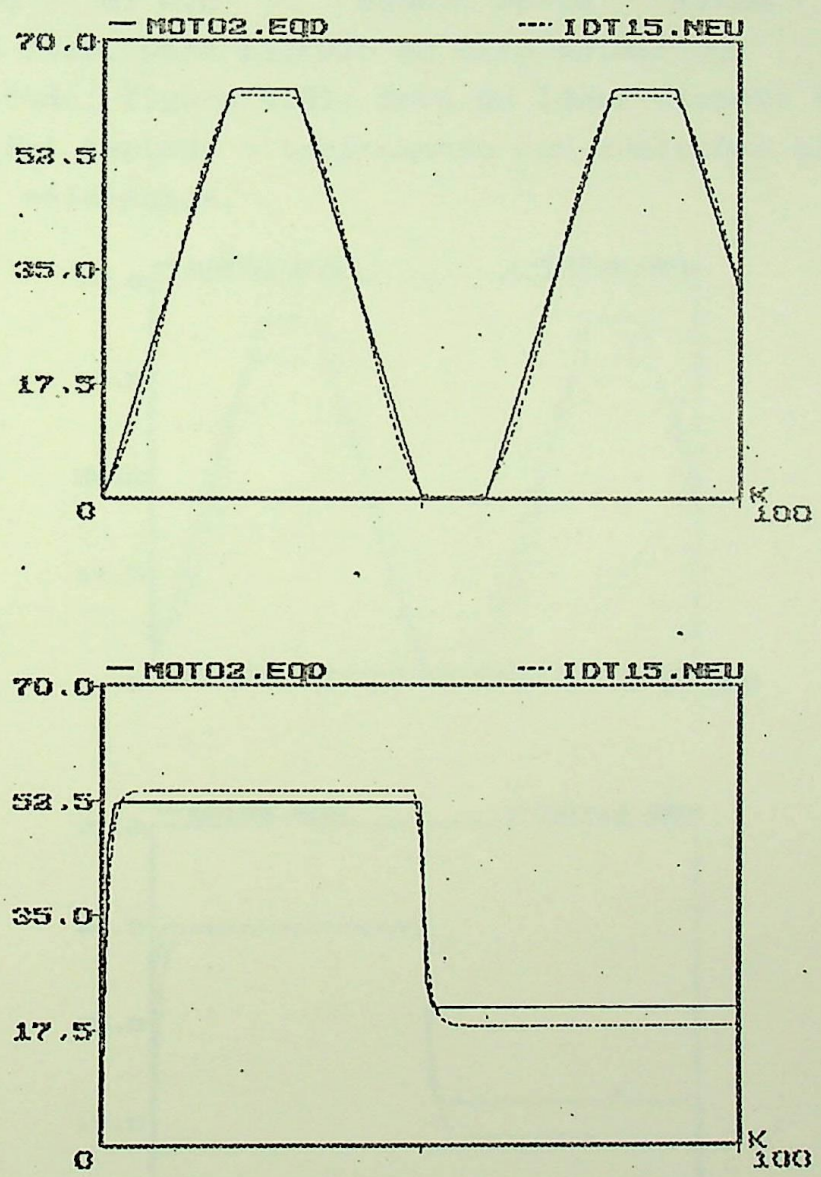


Figura 4.20 - Teste Nº15 - Identificação.

Sinal com referência variando aleatoriamente entre 0 e 6 [V]

OBS.: Dos treinamentos anteriores a melhor resposta foi obtida utilizando uma taxa de aprendizagem η=0,8 e uma taxa de momento α=0,5. Logo, os treinamentos seguintes serão realizados com estes valores.

Teste Nº16

Rede: 3 entradas, 1 camada interna (3 neurônios)
η : 0,8 α: 0,5 Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60
Resultado: figura 4.21; Erro de Identificação Final: 0,0233
Obs: Foi testado o treinamento com variações aleatórias na referência.

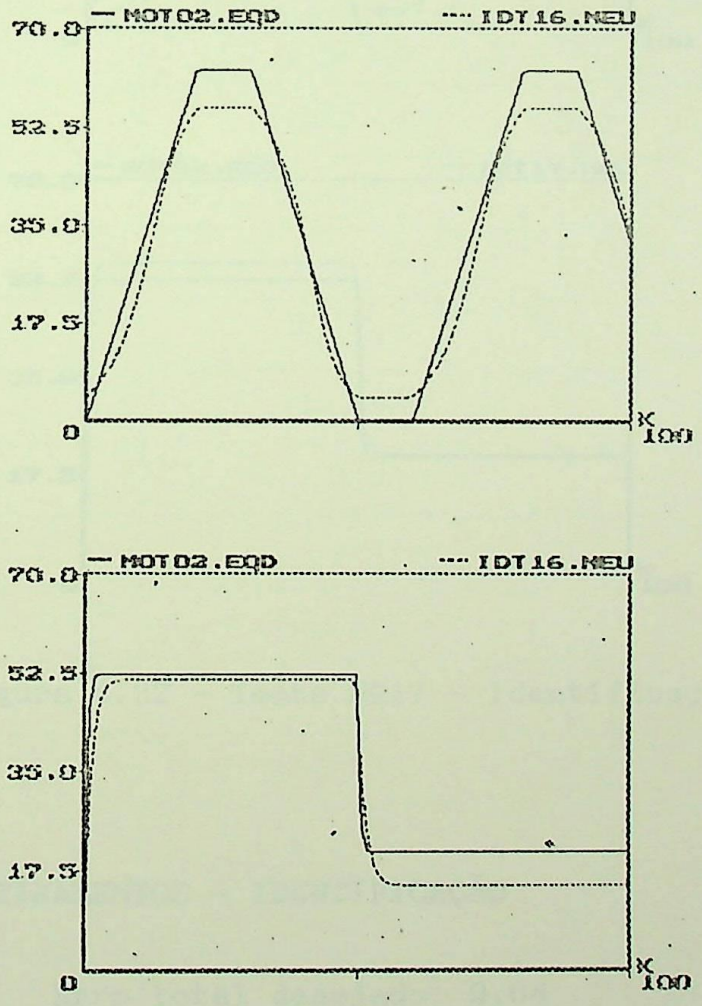


Figura 4.21 - Teste Nº16 - Identificação.

Teste Nº17

Rede: 3 entradas, 1 camada interna (3 neurônios cada)

η : 0,8 α: 0,5 Sinal: Dr=10 Tv=3

N. de iter. para cálculo do erro total: 20

Resultado: figura 4.22; Erro de Identificação Final: 0,0051

Obs: Foi testado o treinamento com variações aleatórias na referência e com um maior número de perturbações.

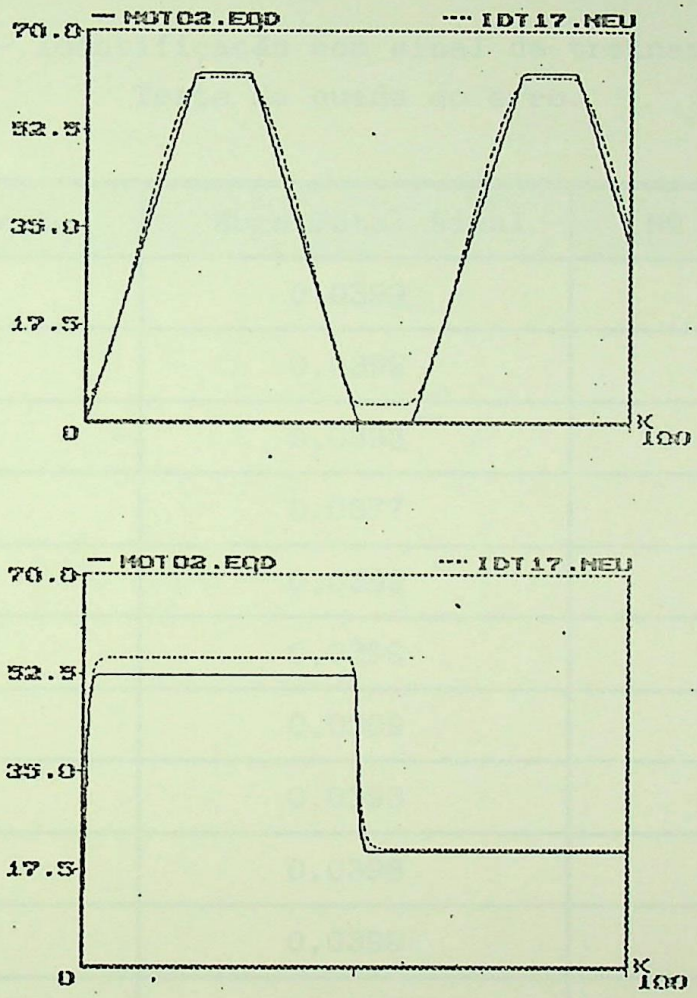


Figura 4.22 - Teste Nº17 - Identificação.

2ª SÉRIE DE TREINAMENTOS - IDENTIFICAÇÃO

Erro total desejado: 0,04

O valor de erro escolhido não necessariamente corresponde

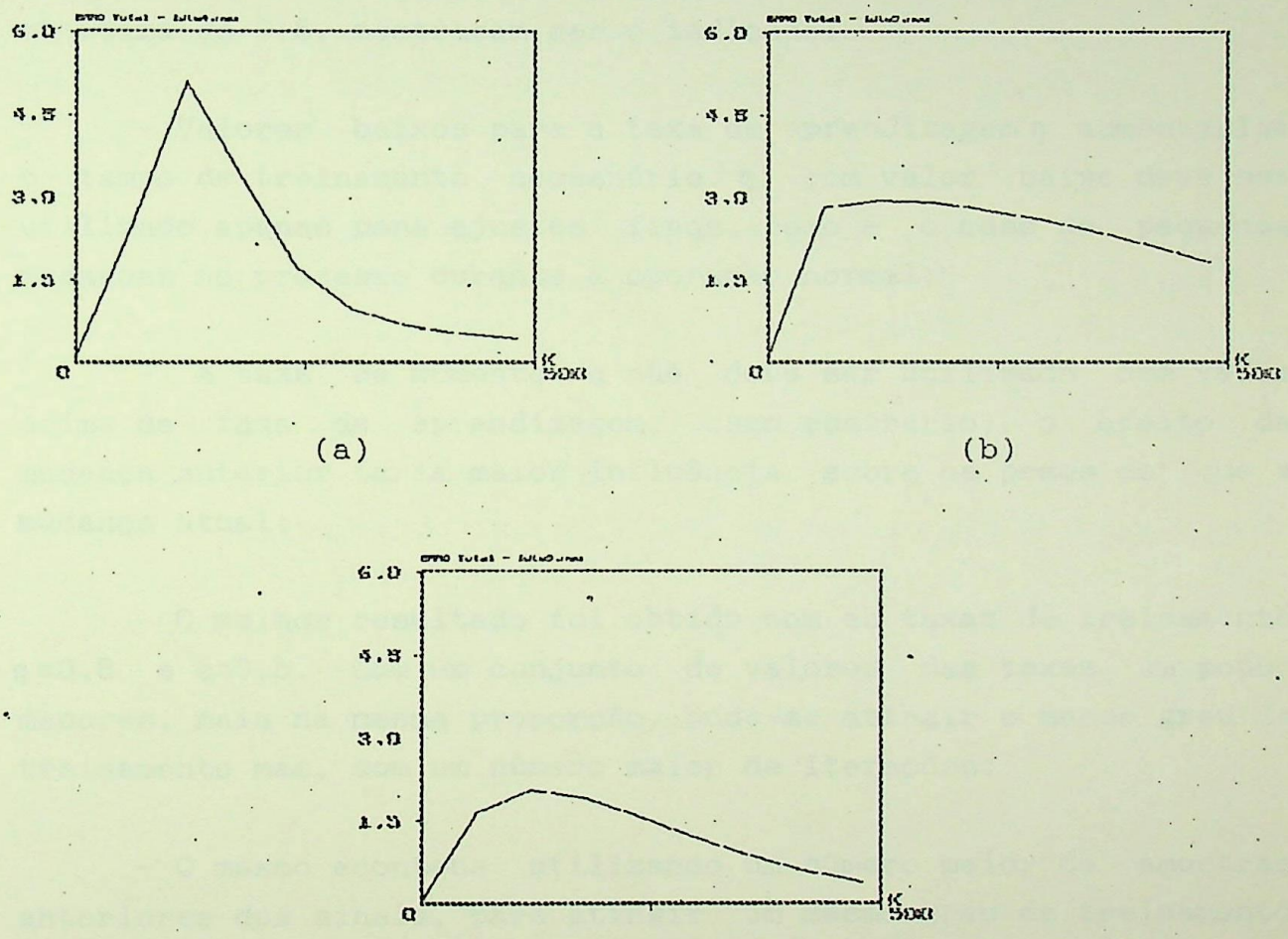
ao valor requerido para um bom grau de treinamento. É um valor que, segundo os testes da 1ª série, deverá ser atingido em todos os casos de teste para fins apenas de comparação do número de iterações necessárias. De qualquer forma será fixado um máximo de 50000 iterações nos casos em que o valor de erro mínimo não seja atingido.

Tabela 4.1 - Identificação com sinal de treinamento fixo - Teste de queda do erro.

Nº do Teste	Erro Total Final	Nº de Iterações
1	0,0399	28.981
2	0,0399	31.981
3	0,0398	18.661
4	0,0377	4.981
5	0,0399	36.181
6	0,0399	19.981
7	0,0399	40.501
8	0,0393	5.581
9	0,0398	12.181
10	0,0399	27.781
11	0,0399	33.241
12	0,0398	12.301
13	0,0394	1.441
14	0,0362	1.741
15	0,0399	2.101

Tabela 4.2 - Identificação com sinal de referência aleatório - Teste de queda do erro.

Nº do Teste	Erro Total Final	Nº de Iterações
16	0,0330	661
17	0,0393	281



(a) (b)

(c)

Figura 4.23 - Comportamento do erro para vários valores das taxas de treinamento.

(a) Teste Nº4

(b) Teste Nº5

(c) Teste Nº9

Análise dos treinamentos e simulações

A partir dos gráficos e dos resultados obtidos podemos tirar algumas conclusões, mostram as respectivas respostas obtidas para os sinais de teste e de treinamento de cada uma das redes:

- Adotamos a rede do teste nº.1 como referência para realizar as comparações;

- Taxas de treinamento com valores médios ou altos, iguais ou acima de 0,5, mostraram ser o indicado;

- Valores baixos para a taxa de aprendizagem η aumentariam o tempo de treinamento necessário. η com valor baixo deve ser utilizado apenas para ajustes finos, como é o caso de pequenas mudanças no processo durante a operação normal;

- A taxa de momento α não deve ser utilizada com valor acima da taxa de aprendizagem, caso contrário, o efeito da mudança anterior teria maior influência sobre os pesos do que a mudança atual;

- O melhor resultado foi obtido com as taxas de treinamento $\eta=0,8$ e $\alpha=0,5$. Com um conjunto de valores das taxas um pouco menores, mais na mesma proporção, pode-se atingir o mesmo grau de treinamento mas, com um número maior de iterações;

- O mesmo acontece utilizando um número maior de amostras anteriores dos sinais, para atingir um mesmo grau de treinamento precisamos de um tempo de treinamento maior;

- Pode-se constatar que aumentando o número de camadas internas obteve-se um melhor resultado, no entanto, isto somente é possível se o período de amostragem disponível permite. Deve-se lembrar que a rede de identificação atua também durante o regime de operação normal;

- Quanto à taxa de subida do sinal de treinamento *Tv*, esta

deverá ser ajustada de acordo com a duração adotada. Seu valor deverá ser tal que a variação da saída do processo não ocorra nem muito rapidamente e nem muito lentamente. Poderíamos neste sentido, utilizar um sinal de treinamento com taxa de variação variável, isto é, teríamos tanto degraus como rampas com taxas diferentes;

- No teste nº.11 pode-se verificar que quanto maior o número de perturbações menor será o tempo de treinamento necessário, visto que a rede terá maior número de informações para um mesmo tempo de observação. Deve-se, no entanto, ter cuidado para que o tempo de variação *Tv* da referência seja o suficintemente grande como para que a rede possa captar a dinâmica do processo;

- Usando variações aleatórias corre-se o risco de variações pequenas na referência, contrário ao sinal de treinamento utilizado nos testes nº.1 a nº.15, que mostrou bons resultados;

- Ampliando a faixa de normalização, isto é, diminuindo a faixa útil da curva sigmoidal (figura 4.24) a rede trabalha apenas em uma parte da região linear da curva. Desta forma a rede pode atingir mais facilmente os valores limites de atuação. Mas, aumentando demais a faixa de normalização, conseqüentemente, a rede utilizará apenas uma pequena faixa da curva, diminuindo com isto a precisão dos valores manuseados;

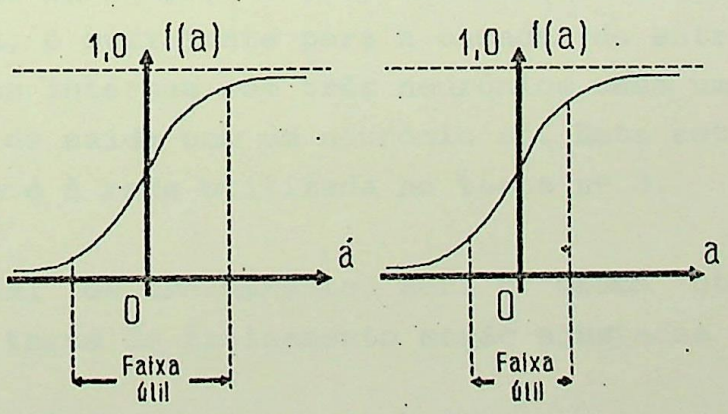


Figura 4.24 - Aumento da faixa de normalização.

- Da segunda série de treinamentos, o teste de velocidade de queda do erro, pode-se confirmar que o tempo de treinamento é reduzido utilizando um maior número de camadas internas, taxas de treinamento altas no começo e uma maior faixa útil da curva sigmoidal.

- A partir dos gráficos da figura 4.23 fica mais claro o efeito dos valores das taxas de treinamento η e α no comportamento do erro global. O gráfico do teste Nº4, figura 4.23(a), mostra um erro alto no início do treinamento, mas decaindo rapidamente. Já, no teste Nº5, figura 4.23(b), com valores menores para η e α, o erro no início é muito menor, mas, este diminue mais lentamente. Na figura 4.23(c), aparece o gráfico do erro do teste Nº9, com as taxas de treinamento ajustadas com os melhores valores.

Conclusão

Levando em consideração as análises que acabaram de ser feitas podemos especificar agora uma estrutura de rede cujo treinamento deve gerar os melhores resultados. Esta rede receberá o nome de *Rede de Identificação Especial*.

Das análises feitas, para o processo em estudo, três neurônios de entradas, isto é, uma amostra anterior da referência e da saída, é suficiente para a camada de entrada. Utilizaremos duas camadas internas com três neurônios cada uma e, como antes, uma camada de saída com um neurônio só. Esta estrutura corresponde exatamente à rede utilizada no teste nº.3.

O sinal de treinamento será o mesmo utilizado no teste nº.3, e as taxas de treinamento serão ajustadas em η=0,8 e α=0.5.

4.3 - TREINAMENTO DO CONTROLADOR - TESTES E RESULTADOS

Para o treinamento do controlador é utilizado o esquema da figura 3.6, para o qual devem ser feitas as mesmas especificações citadas no procedimento de identificação.

Novamente, será feita uma primeira série de treinamentos na qual será fixado um número máximo de iterações para testar o grau de treinamento através do erro total atingido; e uma segunda série, na qual será testada a velocidade de queda do erro total verificando o número de iterações necessário para chegar a um valor mínimo especificado.

Para conferir o comportamento da rede controladora é feita uma simulação da malha de controle com o controlador neuronal acionando o modelo do processo, equação (4.8). Assim, para cada um dos casos da primeira série, após o treinamento, será incluído um gráfico com a resposta da malha ao sinal de treinamento e outro gráfico com a resposta a um sinal de referência de teste formado por dois degraus, de 50 e 20 [V], respectivamente. Considera-se o degrau como o pior caso, ou seja, o tipo de entrada mais crítico. Nos dois gráficos, foi incluído o sinal de referência, com o objetivo de verificar a precisão dos resultados, visto que este representa exatamente a saída desejada.

Para a segunda série de treinamentos, só nos interessa o número total de iterações, o qual será mostrado através de outra tabela.

Devemos lembrar que para o treinamento de cada controlador deve ser utilizada uma das redes já identificadas, escolhida de acordo com o desempenho obtido e levando em conta que as faixas de normalização devem ser iguais. Assim, nos casos em que a rede controladora possua a normalização inicial (0...6;0...70) será utilizada a rede do teste nº9. Nos casos em que a normalização seja diferente será utilizada a rede treinada com a normalização correspondente.

Definição do sinal de treinamento

Levando em consideração as mesmas orientações apresentadas para o sinal de treinamento da identificação, novamente o sinal de referência poderá ser constituído por degraus ou por rampas, como é esperado durante a operação normal. O domínio definido para as entradas e saídas da rede controladora é o mesmo utilizado nas redes de identificação, visto que se trata do mesmo sistema. Assim, a entrada da rede que recebe a referência, as amostras anteriores da própria referência e a realimentação da saída do processo deverão ser normalizadas utilizando a faixa de atuação 0..70, e a saída da rede controladora, assim como as suas respectivas realimentações, deverão ser normalizadas utilizando a faixa 0..6.

O sinal de referência será especificado novamente pelos dois parâmetros mostrados na figura 4.5, a duração de cada valor de referência *Dr* e o tempo de subida ou tempo de variação *Tv*.

Definição da estrutura da rede

Já que a rede com apenas três entradas mostrou ter um comportamento satisfatório, novamente começaremos com uma rede controladora recebendo apenas uma amostra anterior de cada sinal. Outros treinamentos serão realizados variando cada um dos parâmetros de treinamento e a estrutura da rede em termos de número de amostras anteriores captadas pela rede e número de neurônios internos e camadas internas, para análise de desempenho agora a nível de controle.

1ª SÉRIE DE TREINAMENTOS - CONTROLE

Número de iterações: 30.000

Sinal com referência de 0 e 60 [V] variando alternadamente.

Teste Nº1

Rede: 3 entradas, 1 camada interna (3 neurônios)

- ├——> 1 entrada de referência
- ├——> 1 realimentação da saída do processo
- └——> 1 realimentação da saída da rede controladora (entrada do processo)

η : 0,3 α: 0,3 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.25; Erro de Controle Final: 0,0567

Obs: - Este primeiro teste utiliza uma rede com estrutura mínima para servir como ponto de referência ou de comparação para análise de resultados.

- O cálculo do erro total é feito a cada 60 iterações, a duração de duas variações na referência.

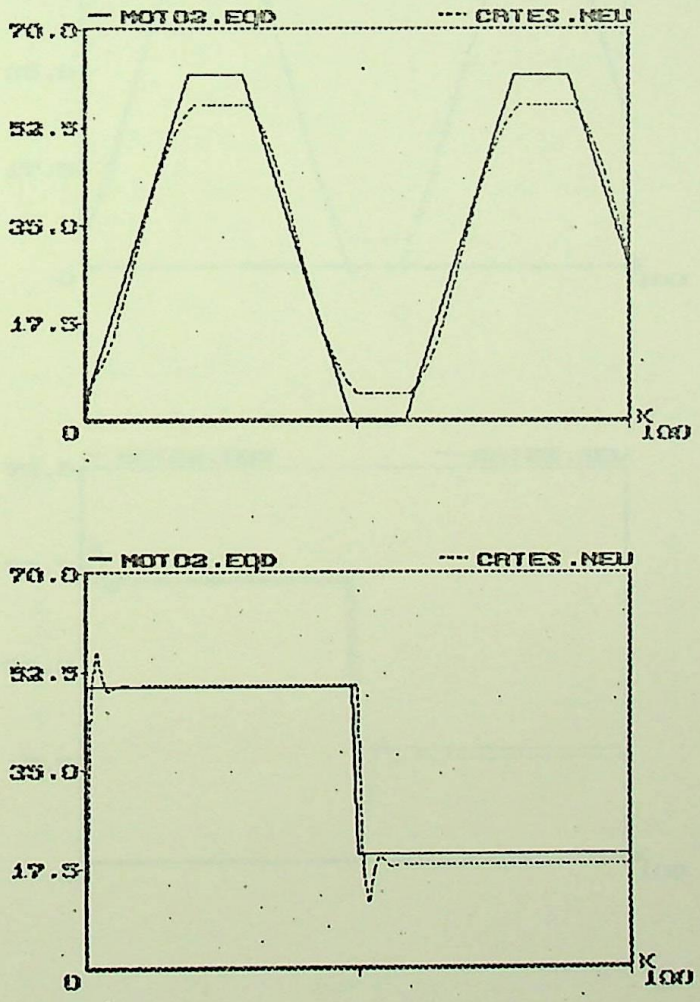


Figura 4.25 - Teste Nº1 - Controle.

Teste Nº2

Rede: 5 entradas, 1 camada interna (5 neurônios)

├——> 1 entrada de referência

├——> 2 amostras anteriores da saída do processo

└——> 2 amostras anteriores da saída da rede controladora (entrada do processo)

η : 0,3 α: 0,3 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.26; Erro de Controle Final: 0,0429

Obs: - Foi aumentado o número de entradas, isto é, de amostras anteriores dos sinais de entrada.

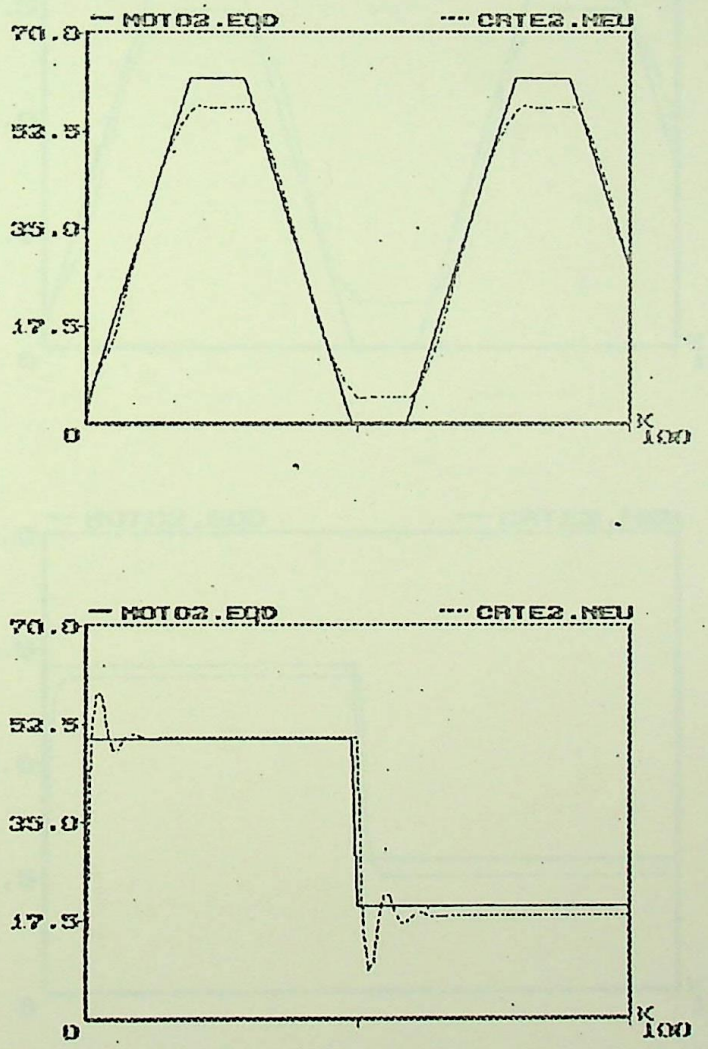


Figura 4.26 - Tese Nº2 - Controle.

Teste Nº3

Rede: 3 entradas, 2 camadas internas (3 neurônios cada)
η : 0,3 α: 0,3
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.27; Erro de Controle Final: 0,1715
Obs: Foi aumentado o número de camadas internas.

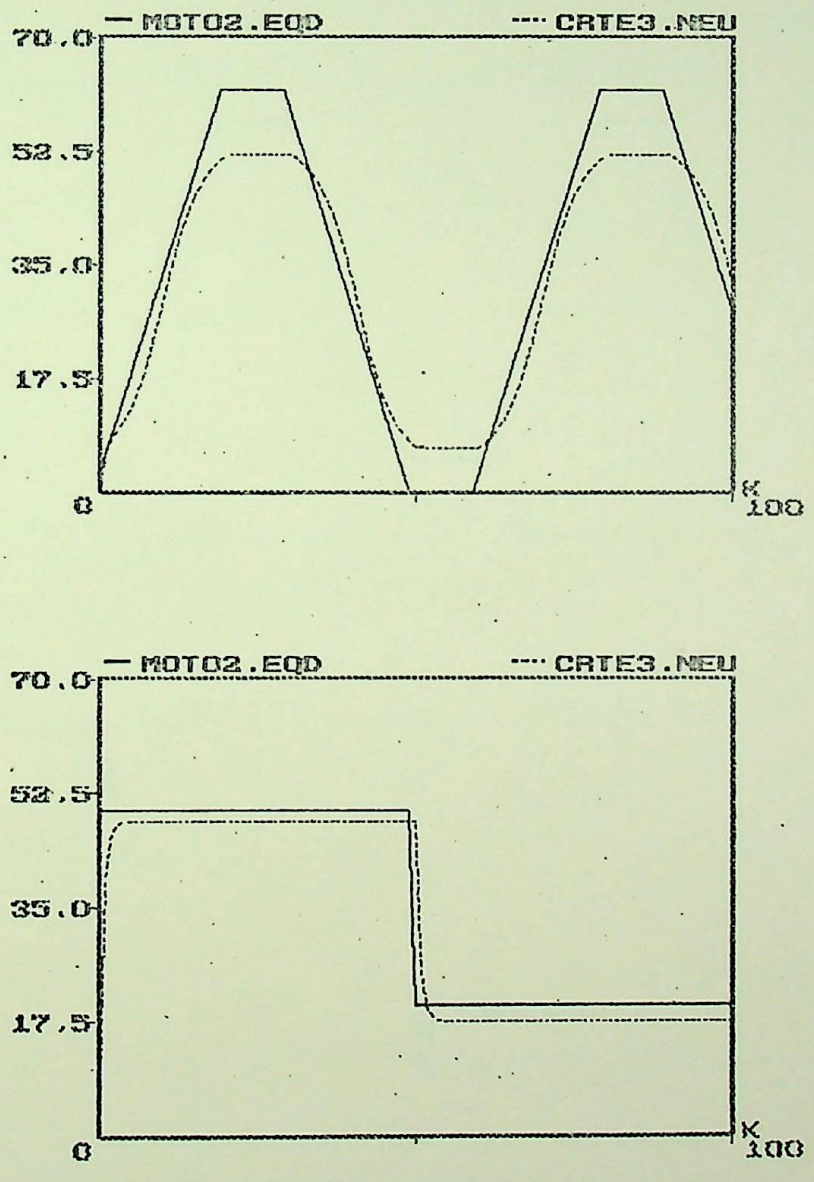


Figura 4.27 - Teste Nº3 - Controle.

Teste Nº4

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,4 α: 0,4

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.28; Erro de Controle Final: 0,0443

Obs: Foi aumentado o valor das taxas de treinamento η e α.

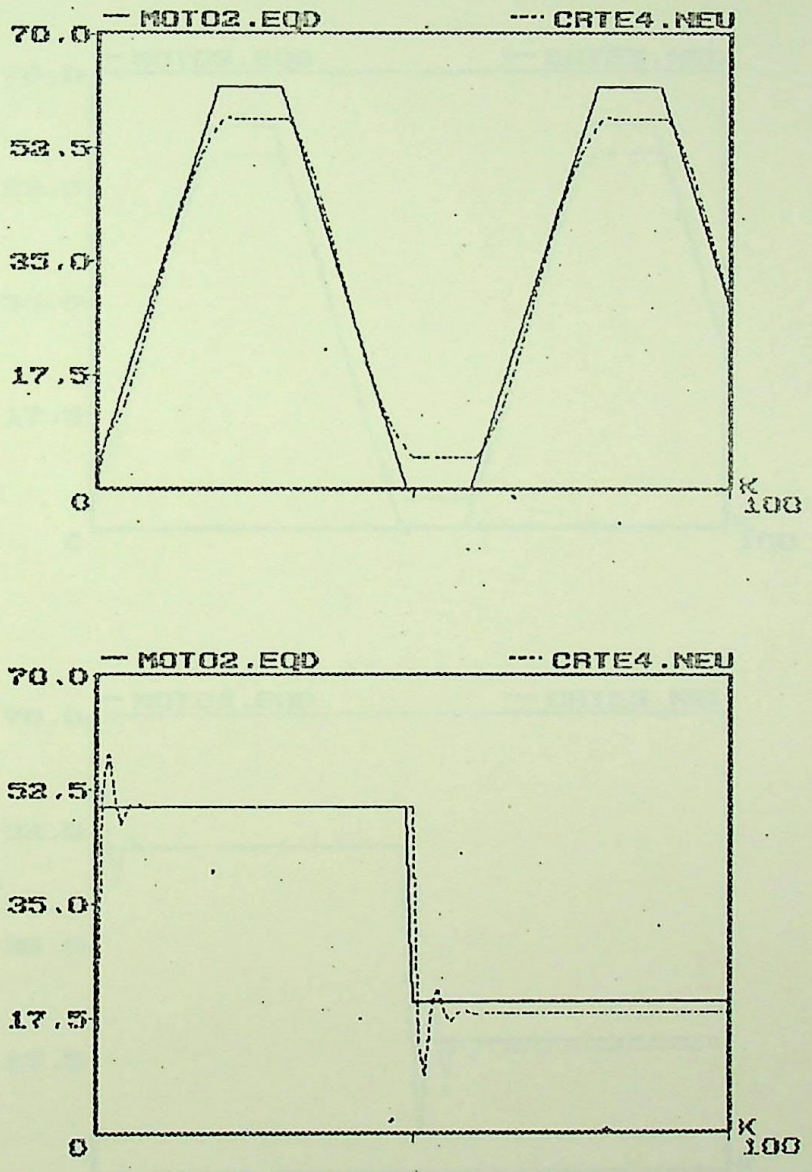


Figura 4.28 - Teste Nº4 - Controle.

Teste Nº5

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,5 α: 0,5

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.29; Erro de Controle Final: 0,0334

Obs: Foi aumentado ainda mais o valor das taxas de treinamento.

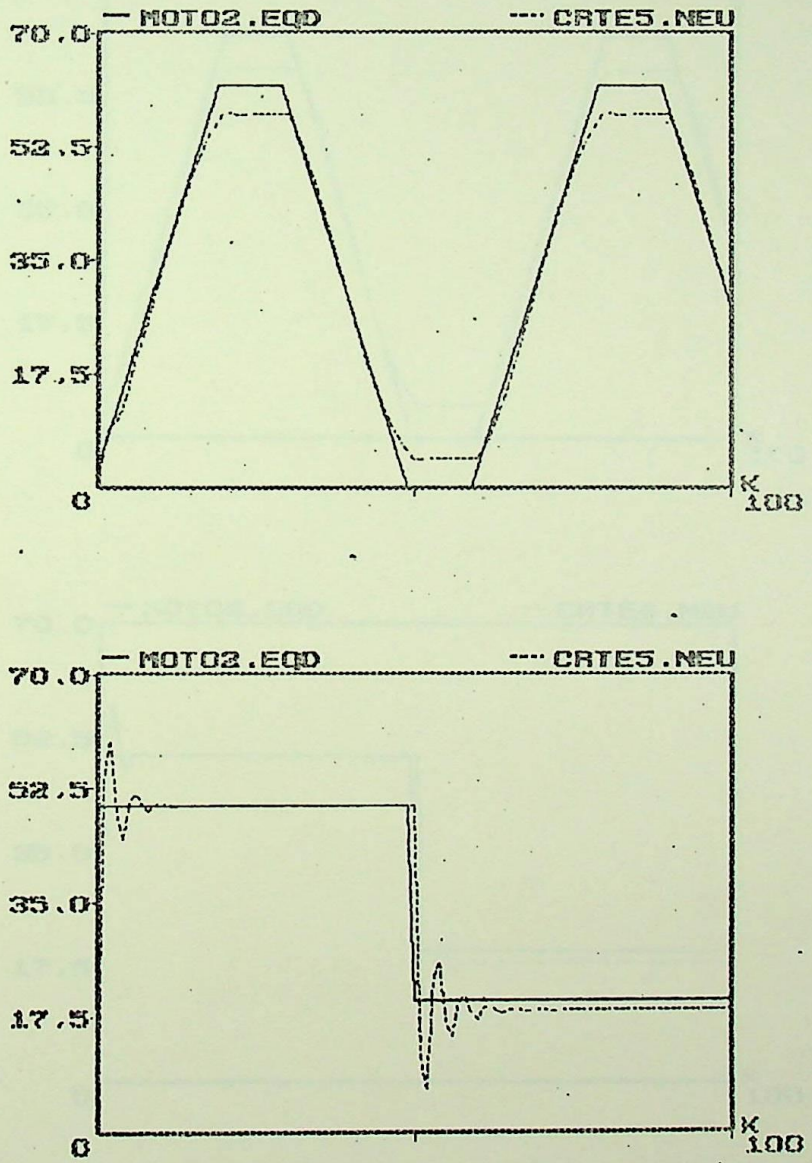


Figura 4.29 - Teste Nº5 - Controle.

Teste Nº6

Rede: 3 entradas, 1 camada interna (3 neurônios)
η : 0,5 α: 0,2
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.30; Erro de Controle Final: 0,0466
Obs: Verificação da ação da taxa de aprendizagem η .

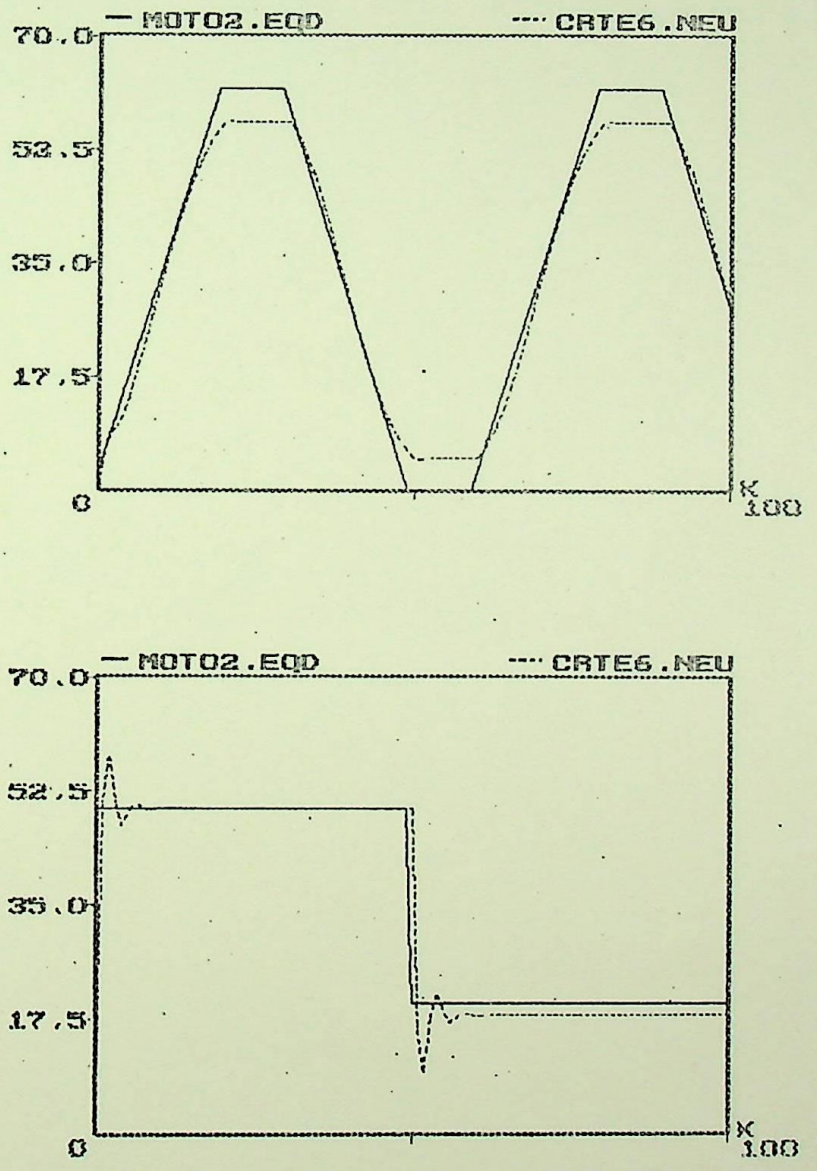


Figura 4.30 - Teste Nº6 - Controle.

Teste Nº7

Rede: 3 entradas, 1 camada interna (3 neurônios)
η : 0,2 α: 0,5
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.31; Erro de Controle Final: 0,0586
Obs: Verificação da ação da taxa de momento α.

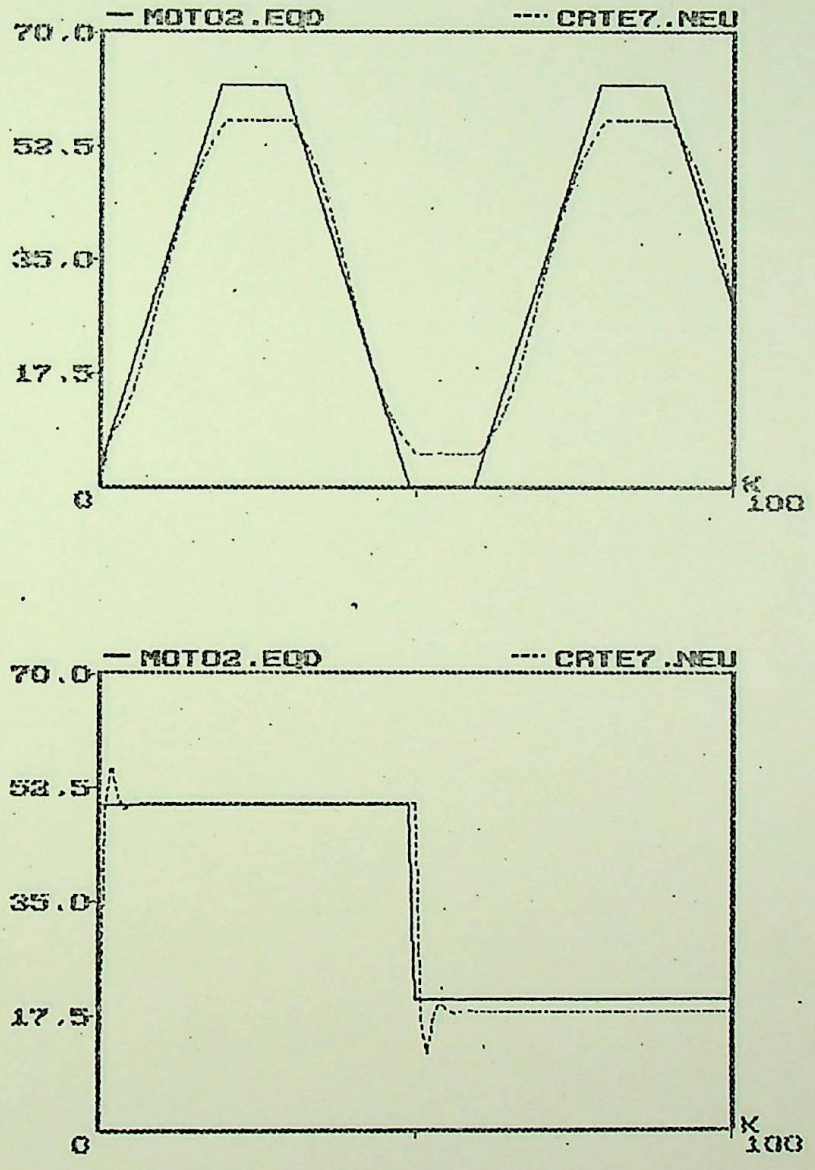


Figura 4.31 - Teste Nº7 - Controle.

Teste Nº8

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,3 α: 0,1

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.32; Erro de Controle Final: 0,0801

Obs: Foi testado outro conjunto de valores para as taxas.

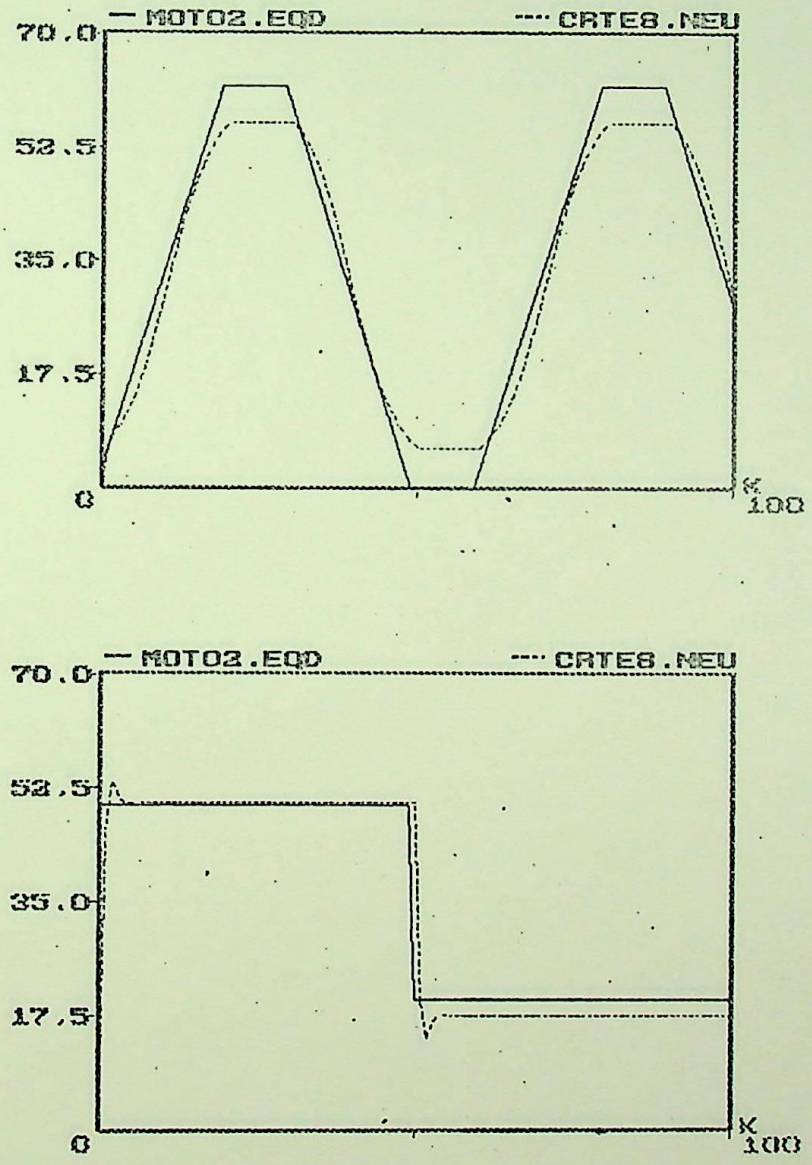


Figura 4.32 - Teste Nº8 - Controle.

Teste Nº9

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,4 α: 0,2

Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.33; Erro de Controle Final: 0,0514

Obs: Foi testado outro conjunto de valores para as taxas.

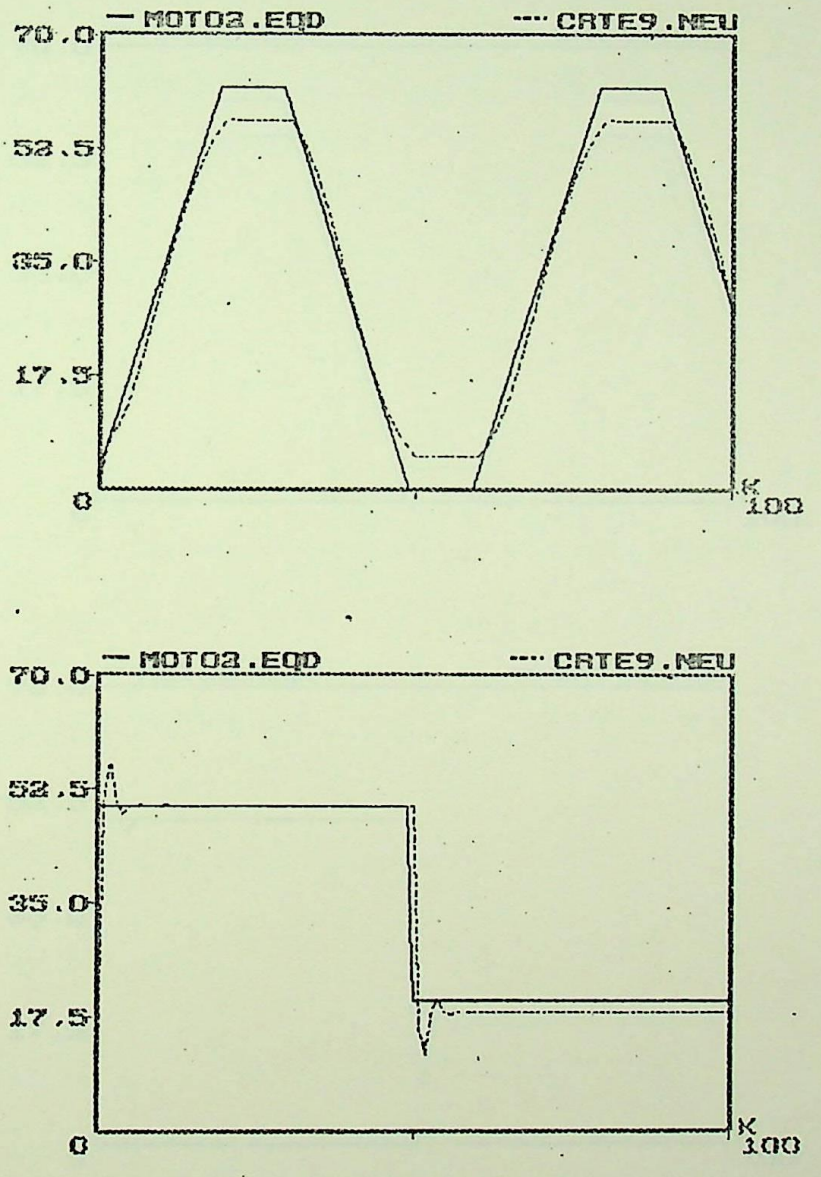


Figura 4.33 - Teste Nº9 - Controle.

Teste Nº10

Rede: 5 entradas, 2 camadas internas (5 neurônios cada)
η : 0,3 α: 0,3
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60

Resultado: figura 4.34; Erro de Controle Final: 0,0455
Obs: Foi aumentado o número de camadas e neurônios internos.

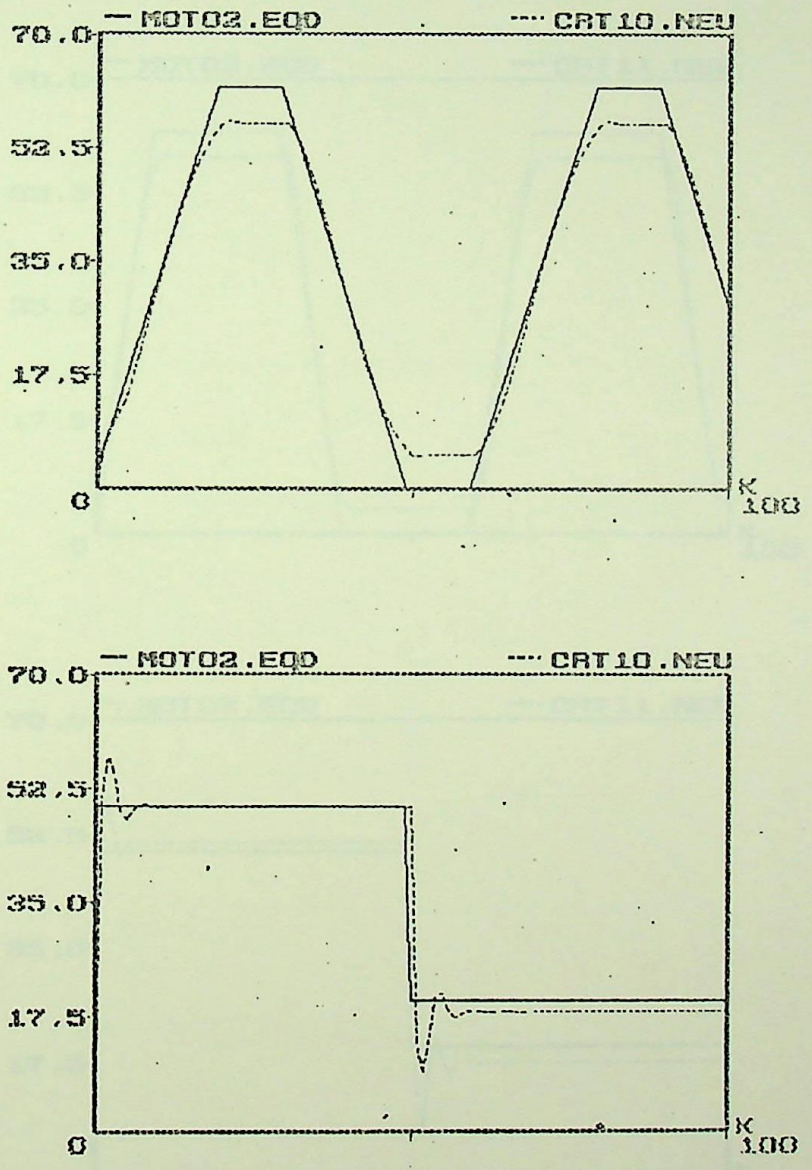


Figura 4.34 - Teste Nº10 - Controle.

Teste Nº11

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,3 α: 0,3

Sinal: Dr=30 Tv=10

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.35; Erro de Controle Final: 0,0617

Obs: Foi aumentado apenas a taxa de variação da referência para aproximá-la mais de um degrau.

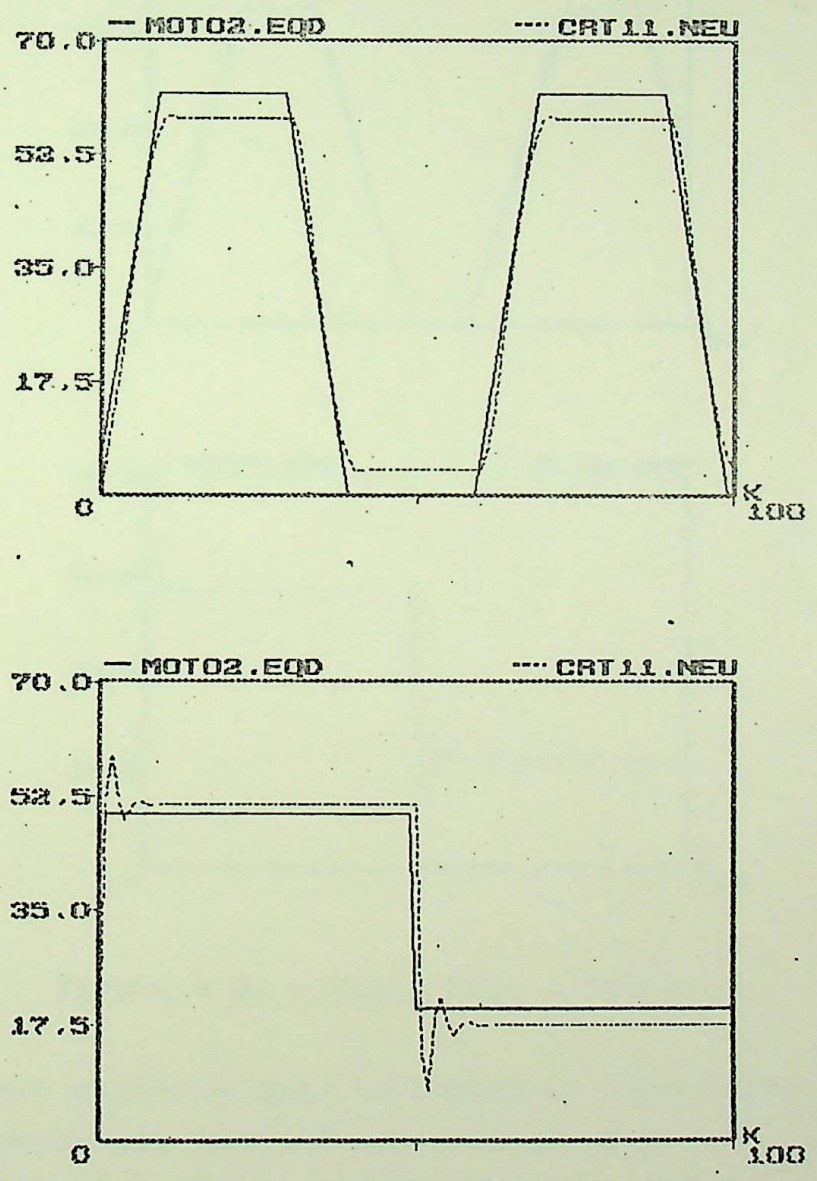


Figura 4.35 - Teste Nº11 - Controle.

Teste Nº12

Rede: 3 entradas, 1 camada interna (5 neurônios)
η : 0,3 α: 0,3
Sinal: Dr=30 Tv=20
N. de iter. para cálculo do erro total: 60
Resultado: figura 4.36; Erro de Controle Final: 0,0428
Obs: Foi aumentado o número de neurônios internos sem aumentar o número de camadas internas.

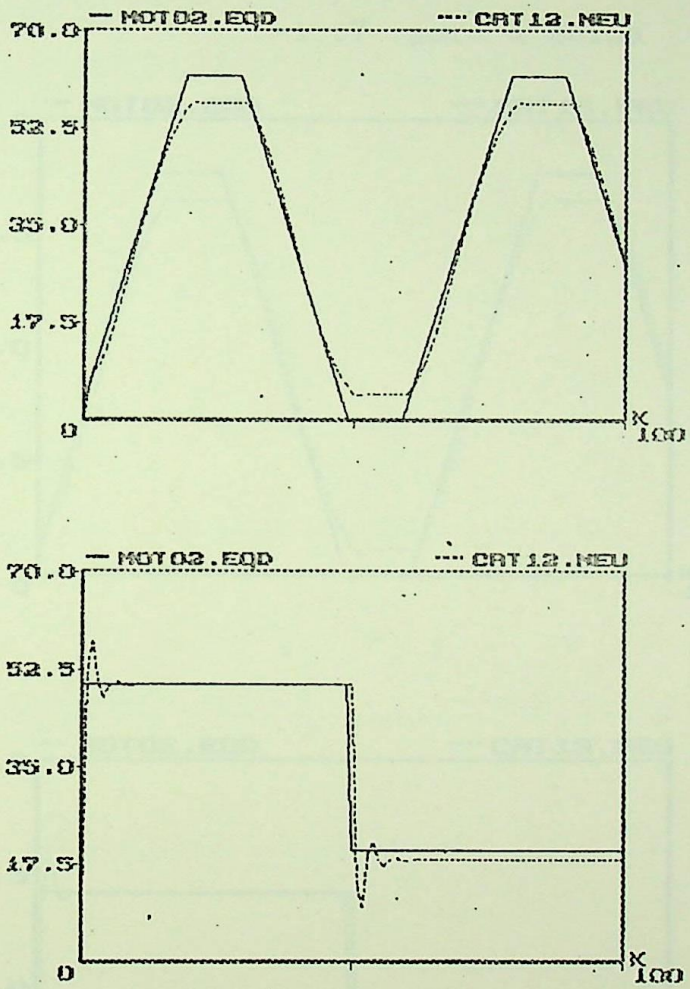


Figura 4.36 - Teste Nº12 - Controle.

OBS.: A melhor resposta dos treinamentos anteriores foi obtida utilizando uma taxa de aprendizagem η=0,5 e uma taxa de momento α=0,2. Logo, os próximos treinamentos serão realizados com estes valores.

Teste Nº13

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,5 α: 0,2 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.37; Erro de Controle Final: 0,0509

Obs: *Mudança na normalização*

Valores anteriores: 0..70 para a referência
0..6 para a saída

Novos valores: -10..80 para a referência
-1..7 para a saída

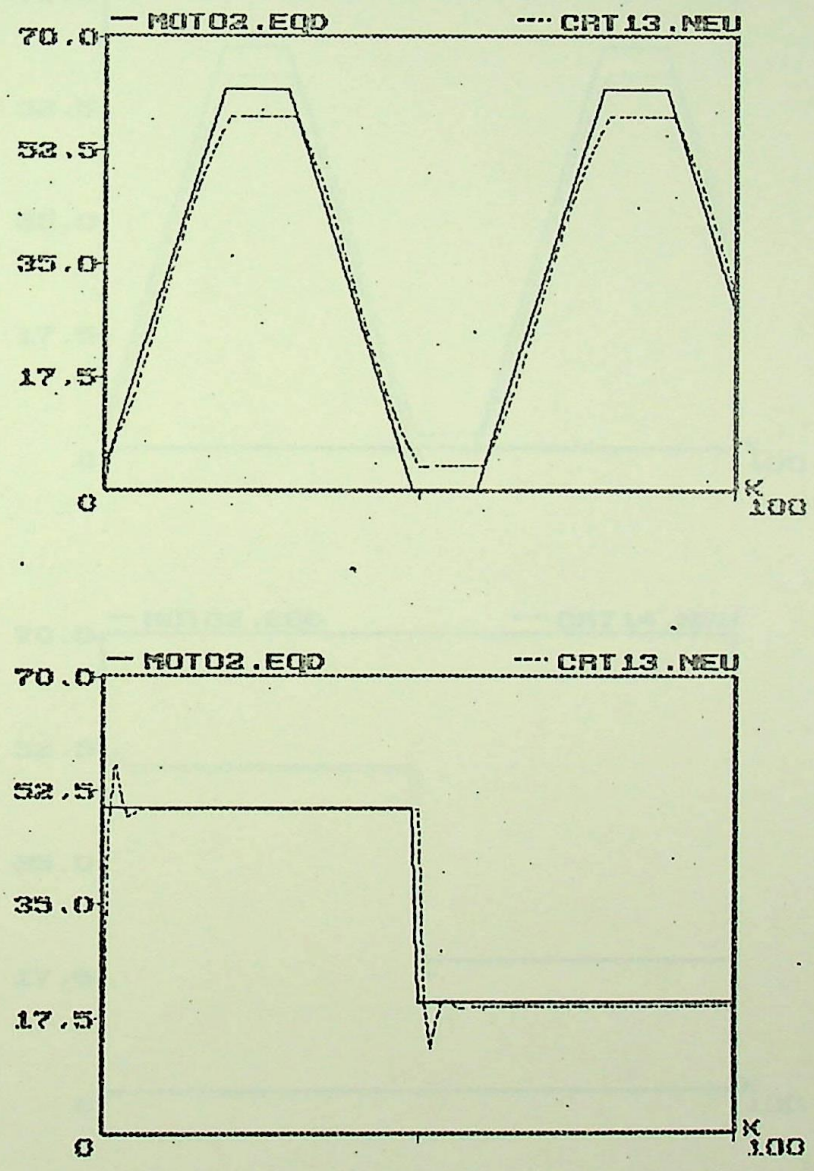


Figura 4.37 - Teste Nº13 - Controle.

Teste Nº14

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,5 α: 0,2 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.38; Erro de Controle Final: 0,0330

Obs: *Mudança na normalização*

Novos valores: -20..90 para a referência

-2..8 para a saída

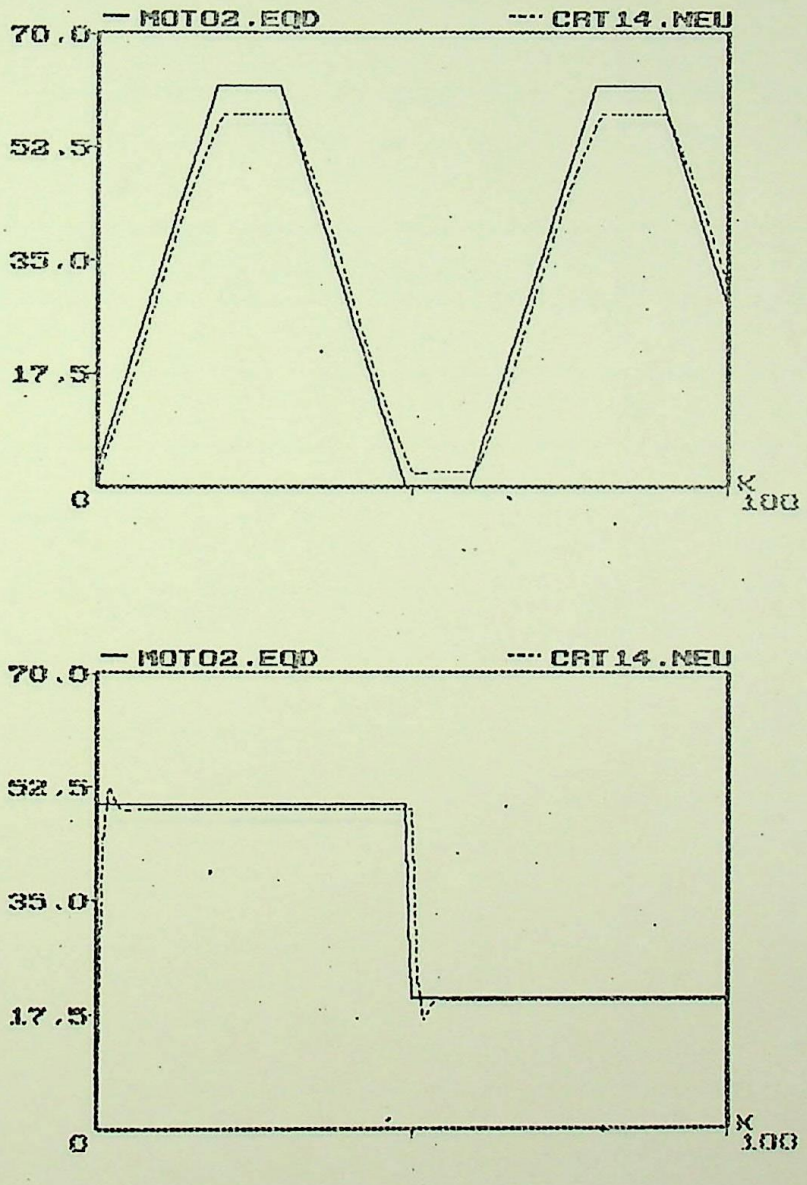


Figura 4.38 - Teste Nº14 - Controle.

Teste Nº15

Rede: 3 entradas, 1 camada interna (3 neurônios)

η : 0,5 α: 0,2 Sinal: Dr=30 Tv=20

N. de iter. para cálculo do erro total: 60

Resultado: figura 4.39; Erro de Controle Final: 0,0296

Obs: *Mudança na normalização.*

Novos valores: -30..100 para a referência

-3..9 para a saída

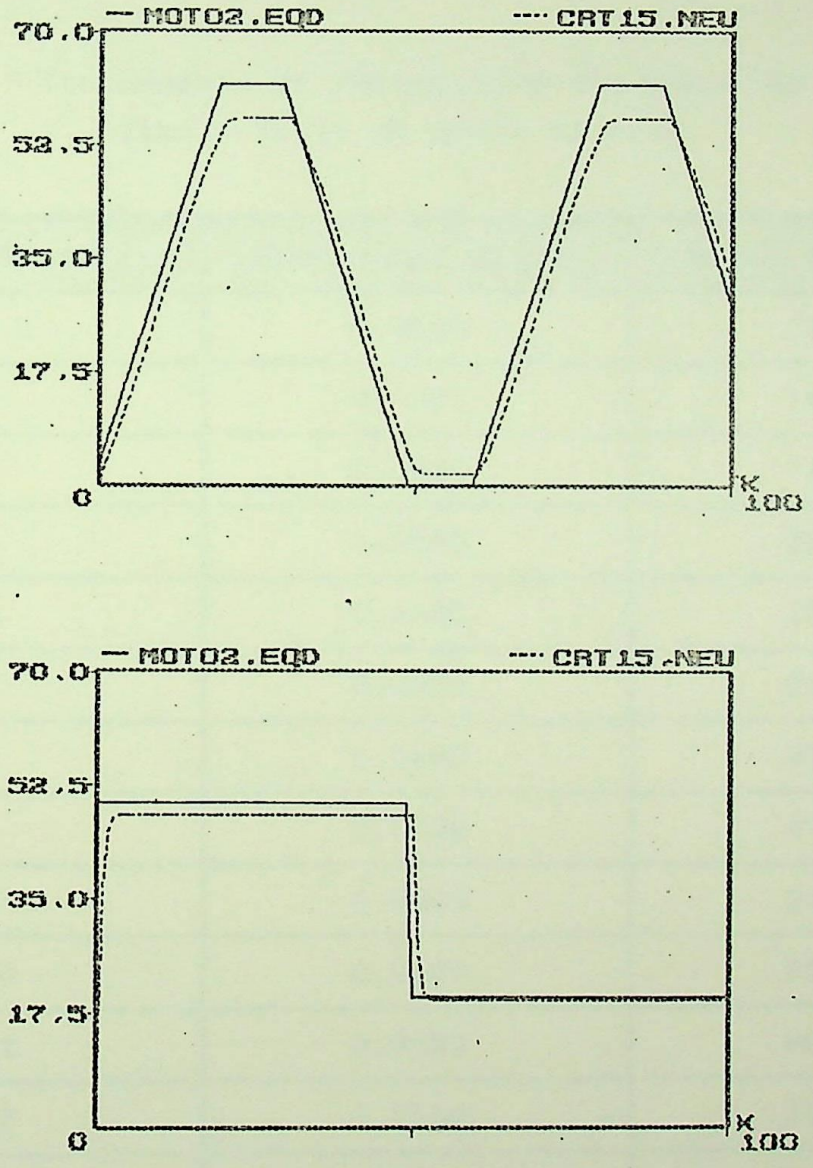


Figura 4.39 - Teste Nº15 - Controle.

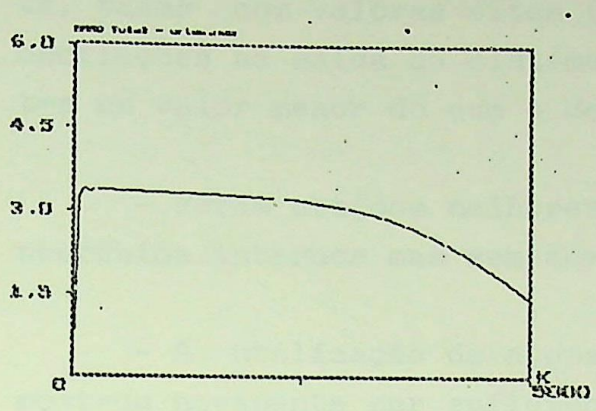
2ª SÉRIE DE TREINAMENTOS - CONTROLE

Erro total desejado: 0,06

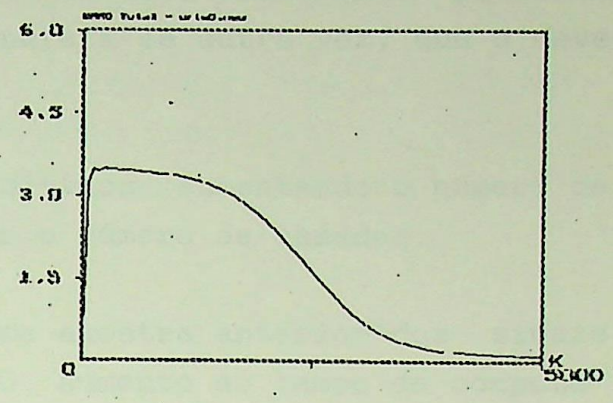
O valor de erro escolhido não necessariamente corresponde ao valor requerido para um bom grau de treinamento. Este valor de erro deverá ser atingido em todos os casos de teste para fins apenas de comparação do número de iterações necessárias. De qualquer forma será fixado um máximo de 50000 iterações nos casos em que o valor de erro mínimo não seja atingido.

Tabela 4.3 - Treinamento do controlador com sinal de treinamento fixo - Teste de queda do erro.

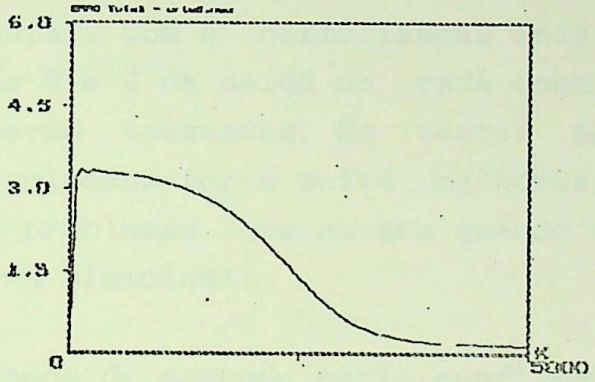
Nº do Teste	Erro Total final	Nº de Iterações
1	0,0599	30.660
2	0,0599	18.840
3	0,0599	39.540
4	0,0599	19.860
5	0,0599	13.260
6	0,0599	20.340
7	0,0599	31.560
8	0,0599	47.700
9	0,0599	24.720
10	0,0599	25.500
11	0,0599	30.840
12	0,0599	19.320
13	0,0599	24.480
14	0,0598	15.420
15	0,0598	11.340



(a)



(b)



(c)

Figura 4.40 - Comportamento do erro para vários valores das taxas de treinamento.
(a) Teste Nº1
(b) Teste Nº5
(c) Teste Nº6

Análise dos treinamentos e simulações

Novamente, podemos tirar algumas conclusões a partir dos treinamentos realizados com os vários controladores:

- A rede do teste nº.1 é novamente utilizada como referência para as comparações;

- Como o treinamento do controlador ocorre de forma indire-

ta, taxas com valores altos (acima de 0,5) facilmente provocam oscilações na saída do sistema. Constata-se outra vez, que α deve ter um valor menor do que o de η ;

- Foram obtidos melhores resultados aumentando o número de neurônios internos mas sem aumentar o número de camadas;

- A utilização de apenas uma amostra anterior dos sinais mostrou novamente ser suficiente. O aumento no tempo de computação não compensa a pequena melhora nos resultados;

- A normalização no treinamento do controlador representou um papel fundamental. Com a normalização inicial (0...6;0...70) os valores limites 0 e 6 na saída da rede controladora tornaram-se difícil de serem acessados. No testes onde a faixa foi aumentada, os resultados foram muito melhores. Cabe lembrar no entanto, sobre os problemas que surgem quando reduz-se demais a faixa útil da curva sigmoidal;

- Os resultados da segunda série confirmam as conclusões do procedimento de identificação; taxas altas no início e faixas de normalização maiores diminuem o tempo de treinamento necessário.

- A partir dos gráficos da figura 4.40, observamos mais claramente o efeito de η e α no ajuste do controlador. Nas figuras 4.40(a) e 4.40(b), vemos como taxas mais altas aceleram a queda do erro global. E, na figura 4.40(c), vemos como taxas de treinamento com valores baixos devem ser utilizadas apenas para ajustes pequenos.

Conclusão

Temos condições agora de especificar uma estrutura de rede que, segundo as análises feitas, deve gerar os melhores resultados para controle. Esta rede terá o nome de *Rede de Controle Especial*, que será treinada utilizando a *Rede de Identificação Especial* e juntas formarão o sistema de controle neuronal que

será utilizado nos testes do item 4.4.

Esta rede será constituída por três neurônios de entrada, uma camada interna com cinco neurônios e uma camada de saída com um neurônio só. Esta estrutura corresponde exatamente à rede utilizada no teste de controle nº.12.

O sinal de treinamento será o mesmo utilizado no teste nº.12, no entanto as taxas de treinamento serão ajustadas em $\eta = 0{,}5$ e $\alpha = 0.2$.

4.4 - ANÁLISE DE DESEMPENHO

Após os procedimentos de identificação e treinamento do controlador, o sistema está pronto para operar sob o esquema da figura 3.4, cujo funcionamento foi explicado no item 3.4. Mas antes, devemos especificar a estratégia de adaptação caso ocorram mudanças no comportamento.

Primeiramente devemos estabelecer uma forma de detectar variações indesejáveis. Isto pode ser feito comparando a variável sob controle e a referência desejada. A comparação deve ser feita independentemente dos valores em questão, o que pode ser feito através do cálculo do *erro relativo*

$$Ep = \frac{Y_{calculado} - Y_{esperado}}{Y_{esperado}} \qquad (4.9)$$

ou do *erro relativo percentual*

$$Ep = \frac{Y_{calculado} - Y_{esperado}}{Y_{esperado}} \times 100 \qquad (4.10)$$

A equação (4.10) pode ser fácilmente computada a cada iteração, comparando o resultado com um valor de tolerância mínima exigida. Dependendo da aplicação, uma tolerância de 5% é

aceitável, no entanto tratando-se de um controle de velocidade, onde a precisão é de suma importância, testaremos o sistema para tolerâncias de até 1%.

Se o erro relativo percentual, calculado em um determinado instante, estiver acima do mínimo desejado, deve-se proceder com o ajuste do sistema.

Dois pontos devem ser analisados: a seqüência de ajuste das redes e o valor das taxas de treinamento. Os dois pontos já foram tratados no item 3.4. De forma resumida, a seqüência de ajuste é a seguinte:

1 - Cálculo do erro de identificação
2 - Propagação reversa do erro de identificação através da rede de identificação
3 - Ajuste dos pesos da rede de identificação
4 - Cálculo do erro de controle
5 - Propagação reversa do erro de controle através da rede de identificação para cálculo do erro equivalente
6 - Propagação reversa do erro equivalente através da rede de controle
7 - Ajuste dos pesos da rede de controle.

Após esta seqüência de ajuste o sistema recebe a próxima amostra do sinal de referência para o qual é realizada a operação normal da malha. Caso o novo erro relativo percentual continue acima do valor desejado a seqüência de ajuste é realizada novamente.

Para evitar que novos ajustes modifiquem de maneira expressiva o comportamento global da rede, esses deverão ocorrer com intensidade proporcional ao valor do erro relativo percentual calculado. É, neste ponto, onde a ação das taxas de treinamento é importante. A intensidade dos ajustes pode ser controlada através dos valores de η e α. Para valores de erro grandes utilizamos taxas com valores altos (entre 0,7 e 1), enquanto que para valores de erro pequenos utilizamos taxas com valores pequenos (0,1 e 0,3).

Como outra sugestão, poderíamos utilizar, no lugar do erro relativo percentual, apenas o erro relativo, e utilizar o próprio valor calculado como sendo o valor das taxas de treinamento, obtendo com isto, um ajuste mais adequado.

Caso sejam utilizados valores diferentes para as taxas de treinamento aplica-se o raciocínio anterior apenas para a taxa de aprendizagem e define-se uma fração desta como valor da taxa de momento.

Nos itens seguintes deste capítulo serão realizados vários testes de verificação de desempenho do sistema de controle neuronal, utilizando as redes especiais, tanto em condições normais de operação como diante de situações adversas, tais como variações de temperatura e ruído.

O programa NEUCONTROL foi desenvolvido especialmente para análise da malha de controle de velocidade do motor CC. Nele estão disponíveis vários tipos de controladores e a capacidade de realização de testes de avaliação, maiores detalhes estão no Apêndice B.

4.4.1 - Mudanças na referência

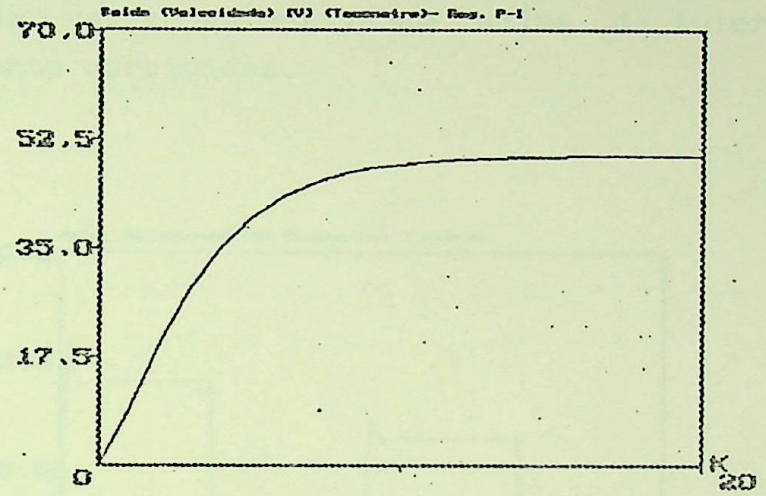
Para analisar a resposta do sistema a variações na referência devemos levar em conta dois aspectos, as caracteristicas da resposta transitória e a precisão da saída.

As caracteristicas mais interessantes da resposta transitória foram apresentadas na figura 1.4 do item 1.1.2.2. Deseja-se uma resposta com o menor "overshoot" (Mo) e os menores tempos de subida (Tr) e de acomodação (Ts).

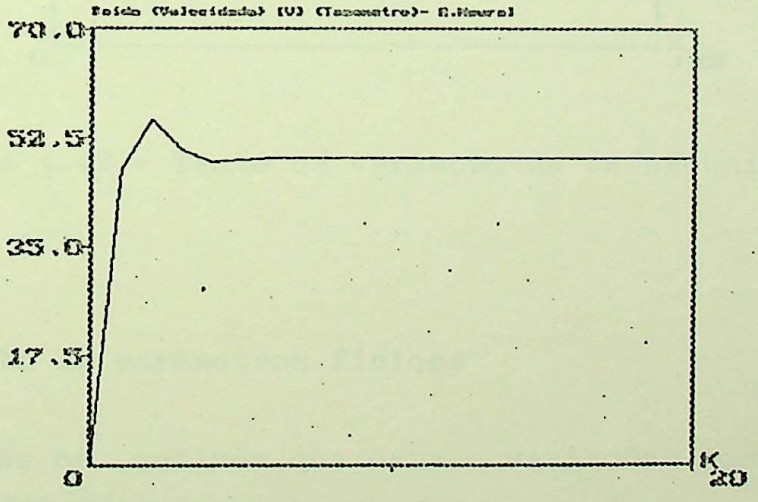
A resposta transitória do sistema neuronal é comparada com a resposta de um sistema com regulador PI, cujos ganhos foram aproximados inicialmente utilizando as regras Ziegler-Nichols mostradas no capítulo 2 e ajustados posteriormente através de

testes de simulação.

Comparando os gráficos da figura 4.41 podemos observar que a resposta da rede possui um sobressinal maior do que o da resposta com o regulador PI, mas em compensação, os tempos de subida e acomodação são bem menores utilizando o controlador neuronal.



(a)



(b)

Figura 4.41 - Resposta ao degrau.
(a) Regulador PI;
(b) Controlador neuronal.

Devemos lembrar que o sobressinal da resposta transitória do sistema neuronal está ligado diretamente ao valor das taxas de treinamento, logo, podemos esperar que com a utilização de taxas variáveis o sobressinal possa ser reduzido.

Outro teste é mostrado na figura 4.42, onde o sistema é submetido a vários valores de referência para avaliar a precisão da saída. Em alguns valores de referência, há uma pequena variação entre o valor esperado e o valor real, no entanto se estas pequenas variações estiverem acima da tolerência de 1% serão devidamente corrigidas.

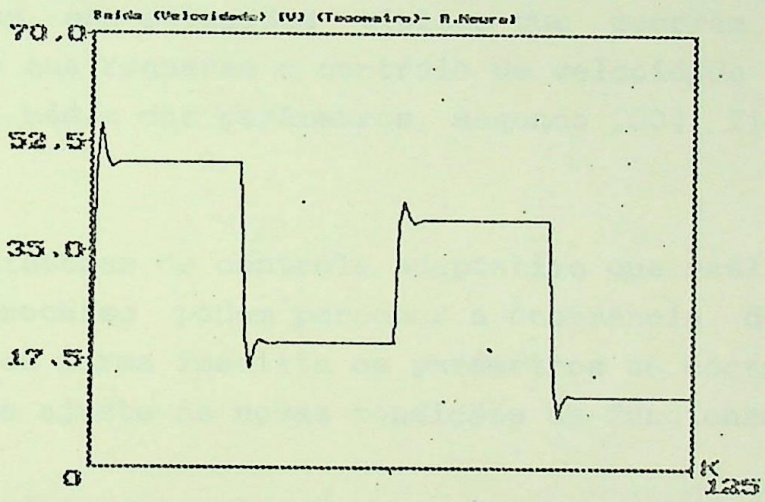


Figura 4.42 - Teste de variação da referência.

4.4.2 - Variação de parâmetros físicos

Vários são os motivos que causam variações nos parâmetros físicos de um determinado sistema.

O comportamento dinâmico do motor CC, por exemplo, varia com o momento de inércia e com o coeficiente de fricção da carga, que por sua vez varia de acordo com a velocidade. Se estas variações são levadas em consideração durante a modelagem do sistema o modelo tornaria-se não-linear, e sabemos que sistemas

não-lineares são difíceis de serem tratados matemáticamente.

O processo de medição da resistência de armadura da máquina, Ra, para utilização em modelos, utiliza o método clássico do voltímetro e o amperímetro, através de testes durante o período de operação e para valores de corrente nominais. No entanto, as condições ambientais podem causar variações nos resultados desta medida.

A ponte tiristorizada introduz perturbações estruturais no sistema de controle quando o ângulo de disparo varia. Isto afeta tanto o ganho equivalente da ponte como a constante de tempo de processo.

Estas são situações típicas que ocorrem na maioria das aplicações que requerem o controle de velocidade e/ou de posição. A variação média dos parâmetros, segundo [50], fica em torno de 3 por cento.

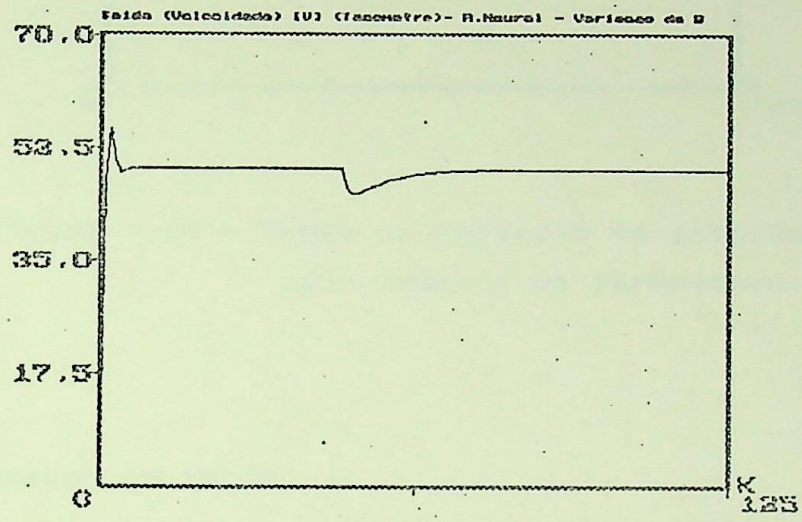
Os sistemas de controle adaptativo que realizam a identificação do processo podem perceber a ocorrência destas mudanças e atualizar de forma imediata os parâmetros do controlador afim de que este se ajuste às novas condições de funcionamento.

As redes neuronais, de uma forma semelhante, podem acompanhar tais mudanças também. Durante o regime de operação normal, o sistema neuronal da 3.4 deverá modificar o comportamento das redes para que estas acompanhem as mudanças ocorridas, seja qual for a sua natureza.

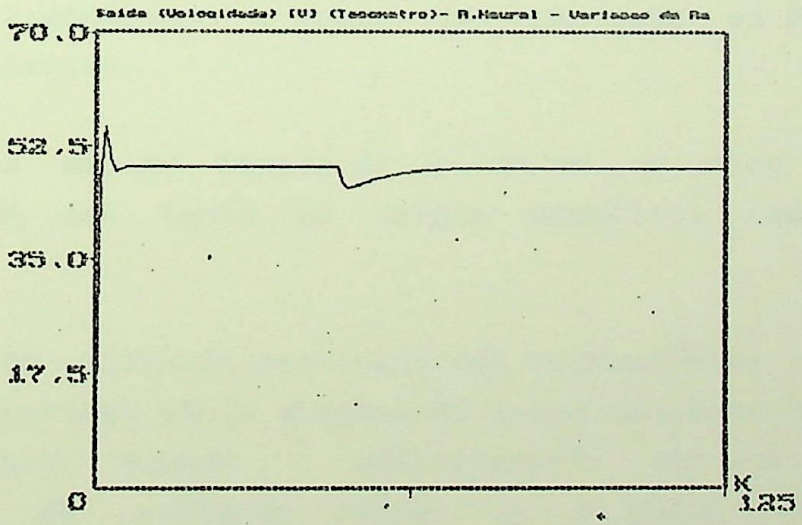
Para verificar o comportamento do sistema diante de algumas das situações apresentadas neste item, o software utilizado foi preparado para simular duas delas. Na figura 4.43(a) é simulada uma variação no coeficiente de fricção B em torno de 50%, e na figura 4.43(b) uma variação na resistência de armadura da máquina.

Nos dois casos, o sistema consegue responder corretamente, no entanto a velocidade de resposta poderia ser ainda maior.

Para verificar se o sistema sofre alguma queda de desempenho após reagir a uma mudança de parâmetros, o mesmo teste da figura 4.43(a) é repetido, e após este, são feitas algumas variações na referência, o resultado é mostrado na figura 4.44. Notamos uma mudança no comportamento do sistema após o ajuste das redes, isto quer dizer que, dependendo da técnica de ajuste adotada, esta pode prejudicar ou melhorar o desempenho do sistema.



(a)



(b)

Figura 4.43 - Teste de variação de parâmetros.
(a) Variação do coeficiente de fricção B;
(b) Variação da resistência de armadura Ra.

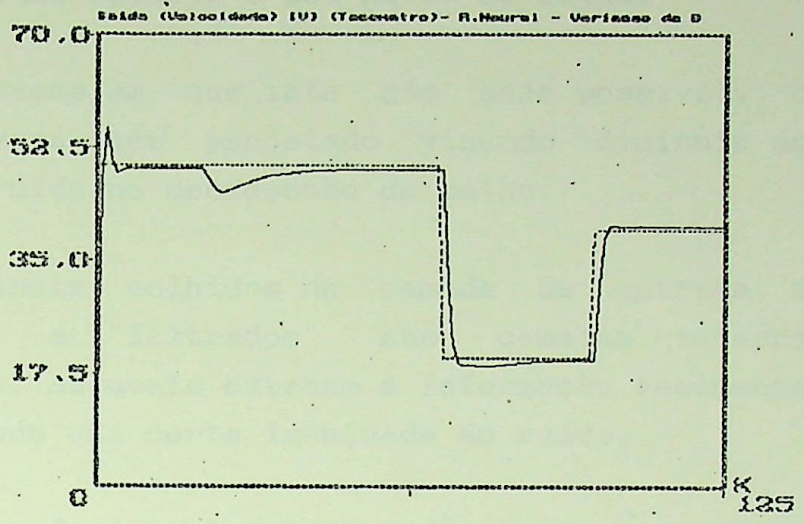


Figura 4.44 - Teste de variação da referência após mudança de parâmetros.

4.4.3 - Presença de ruído

Todos os sistemas de controle físicos estão sujeitos a algum tipo de sinais estranhos ou ruído durante a operação. Exemplos destes sinais são a tensão térmica de ruído em amplificadores eletrônicos e o ruído de comutação ou de escova em motores elétricos.

Muitas são as fontes de ruído em um motor elétrico, as quais podem ser tanto de origem magnética como de origem mecânica.

O ruído em sinais realimentados provenientes de transdutores é um problema sério enfrentado pelos sistemas realimentados, principalmente, quando o comportamento do sistema depende diretamente da validade deles. Se o ruído atingir níveis consideráveis o sinal sofrerá uma distorção tal que o desempenho da malha poderá ser comprometido.

A adequada compreensão dos mecanismos que dão origem a ruídos em um motor é essencial para permitir alterações no

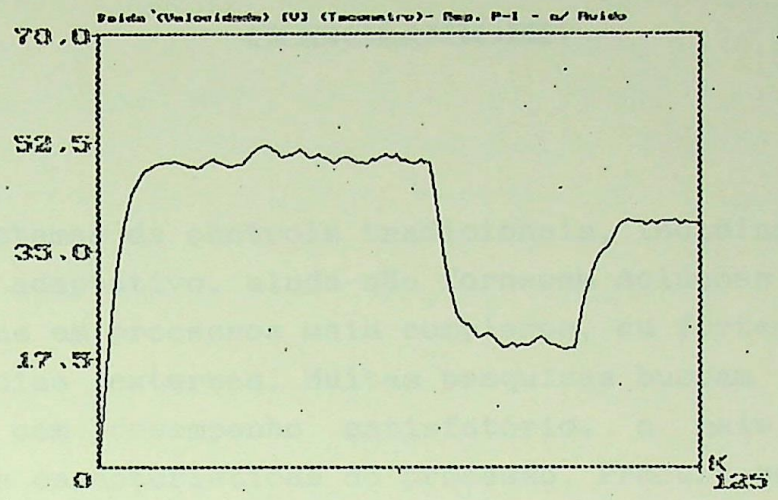
projeto visando reduzir o seu nível de ruído.

Nos casos em que isto não seja possível, o sistema de controle deverá ser projetado visando diminuir ao máximo os efeitos do ruído no desempenho da malha.

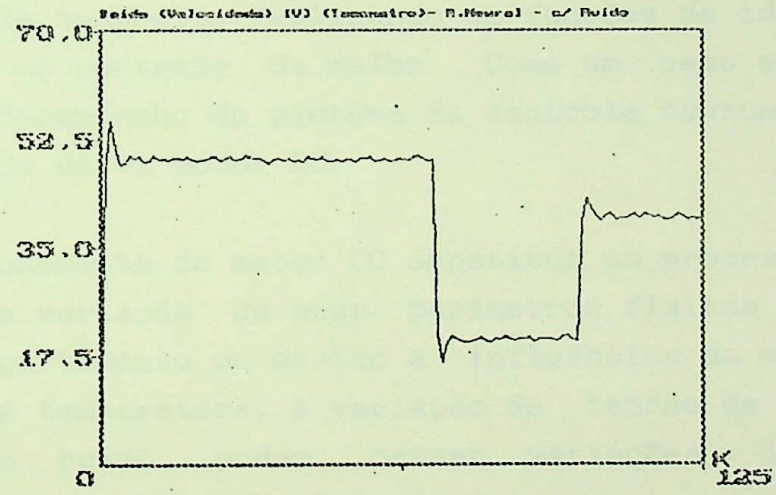
Os sinais colhidos na camada de entrada da rede são processados e "filtrados" nas camadas internas da rede controladora, as quais extraem a informação realmente importante. Isto dá à rede uma certa imunidade ao ruído.

Para conferir tal capacidade, será acrescentado um ruído aleatório na faixa entre 0 e 9 [V] no sinal realimentado pelo transdutor, o que pode representar uma variação de até 20%.

A resposta do sistema neuronal é comparada com a resposta, diante do mesmo tipo de ruído, da malha com regulador PI do item 4.4.1. Os resultados aparecem em termos de gráficos nas figuras 4.45(a) com o regulador PI e 4.45(b) com o controlador neuronal, onde a vantagem do controlador neuronal em ambientes com muito ruído torna-se extremamente clara. A maioria dos controladores PI e sistemas de controle adaptativo perdem toda sua eficácia nestes casos.



(a)



(b)

Figura 4.45 - Teste na presença de ruído.
(a) Regulador PI;
(b) Controlador neuronal.

CONCLUSÕES

Os sistemas de controle tradicionais, incluindo os sistemas de controle adaptativo, ainda não fornecem soluções completamente satisfatórias em processos mais complexos, ou fortemente afetados por influências externas. Muitas pesquisas buscam uma estrutura flexível, com desempenho satisfatório, o mais independente possível das características do processo. Procura-se um estrutura capaz de ajustar-se e atualizar-se de forma automática, e cujo desempenho se mantenha mesmo diante de situações anormais de funcionamento.

Foi analisado o comportamento das redes neuronais dentro de uma malha de controle, realizando as funções de identificação do processo e de controle da malha. Como um caso específico, foi estudado o desempenho do sistema de controle neuronal no controle de velocidade de um motor CC.

O acionamento do motor CC constitui um processo não-linear, que sofre a variação de seus parâmetros físicos devido ao seu próprio comportamento ou devido a influências do meio em que se encontra. A temperatura, a variação da tensão de alimentação da rede e o ruído podem causar variações inesperadas no comportamento da malha de controle.

O sistema de controle deve ser capaz de superar tais inconvenientes, mantendo o desempenho do sistema em um nível satisfatório. Para isto, foram utilizadas as redes neuronais nas funções principais do sistema de controle adaptativo, visando as suas características mais destacadas, entre elas, o paralelismo, a capacidade de aprendizagem e a tolerância a falhas.

Foi adotada a rede do tipo Multicamadas devido ao bom desempenho mostrado no mapeamento de funções lineares e não-lineares. O algoritmo de treinamento utilizado foi o *back-propagation*.

Para a utilização da rede multicamadas em aplicações de tempo real, o algoritmo de treinamento sofreu algumas modificações. O erro médio global, que antes era calculado sobre todo o conjunto de padrões de treinamento disponíveis, agora é calculado sob um determinado intervalo de tempo ou número de iterações. Não dispõe-se de antemão de um conjunto de padrões fixo, estes são formados no decorrer do processo de treinamento.

O sistema de controle neuronal é constituído pelo esquema da figura 3.4 e por dois esquemas derivados deste, o esquema de identificação, mostrado na figura 3.2, e o esquema de ajuste do controlador, mostrado na figura 3.6.

Para o treinamento, a rede recebe amostras atuais e atrasadas dos sinais importantes. No caso do procedimento de identificação, a rede recebe amostras do sinal de referência e da saída do processo. Pretende-se, com isto, que a rede assimile a dinâmica do processo, criando um modelo que possa ser utilizado posteriormente para otimização do processo ou para o ajuste de um controlador.

O ajuste de uma rede no papel do controlador é realizado através de um processo de treinamento indireto. O erro na saída do esquema da figura 3.6 é propagado de forma reversa primeiro através das camadas de uma outra rede, que foi utilizada para identificar o processo. Com isto, obtém-se um erro equivalente na saída da rede controladora, com o qual pode ser realizado o treinamento da mesma.

O ajuste indireto do controlador dá mais precisão e confiabilidade ao sistema, mas este precisa de um tempo de computação muito maior do que nos sistemas de controle direto. No entanto, o treinamento do controlador ocorre sem participação do processo, necessita-se apenas um grau de treinamento aceitável da rede de identificação.

Foram observados, através de simulações, o comportamento do treinamento sob diversas condições. Os parâmetros de treinamento foram variados de diversas maneiras para determinar os valores

que levariam a rede ao melhor grau de aprendizado, ou seja, a um erro global mínimo e a uma precisão aceitável na saída. Foram analisados vários tipos de sinais de referência para treinamento, constituídos por degraus ou rampas. Estes sinais são esperados durante a operação normal do processo.

Foi estudada a influência do valor das taxas de treinamento η e α no resultado final do treinamento, tanto no procedimento de identificação como no ajuste do controlador.

No final das simulações e dos testes realizados, foi especificada uma estrutura de rede especial para identificação do processo e uma outra para o papel do controlador, as quais foram treinadas sob condições de aprendizado consideradas como as melhores.

No último capítulo, foram realizados vários tipos de teste para avaliar o desempenho do sistema desenvolvido sob condições de operação normais e anormais.

O sistema mostrou uma boa precisão no acompanhamento da referência. Foi mostrada a sua capacidade de adaptabilidade através do ajuste das redes após pequenas variações nos parâmetros do processo, diante das quais a resposta pode ser considerada um pouco lenta. A rede enfrenta problemas com variações de parâmetros muito acentuadas. Isto deve-se à capacidade de aprendizagem da rede, da mesma forma como a rede consegue memorizar um determinado comportamento, o treinamento excessivo sob um determinado padrão pode provocar a perda de conhecimento, comprometendo assim o comportamento global da rede em regime de operação normal. Técnicas para acelerar a recuperação do sistema sem prejudicar o desempenho serão objeto de futuros estudos.

O melhor desempenho do sistema foi observado na presença de ruído no sinal de realimentação vindo do transdutor. A resposta do sistema neuronal foi comparada com a de um sistema de controle utilizando o regulador PI, mostrando resultados melhores que ele. Comprovou-se que as camadas internas da rede conseguem eliminar

quase que totalmente o ruído presente nos sinais de entrada. Na análise, foi inserido um ruído aleatório de até 20%, para o qual a rede mostrou um resposta altamente satisfatória.

Este resultado pode servir como orientação sobre a melhor utilização das redes neuronais em sistemas de controle. Sistemas com variações pequenas e com alto nível de ruído se encaixam perfeitamente dentro dos requisitos para uma boa solução utilizando as redes neuronais.

A aplicação das Redes Neuronais na solução de problemas de engenharia elétrica e de controle ainda está em sua fase inicial, porém, sua expansão ocorre de forma acelerada. Hoje, são poucas as aplicações colocadas em prática, no entanto, o potencial observado é muito grande. Para um avanço ainda maior nas aplicações em tempo real, são necessários novos desenvolvimentos dos algoritmos de treinamento e das técnicas de implementação VLSI.

Estudos futuros poderão analisar o desempenho de outros tipos de topologias de redes, assim como, outros algoritmos de treinamento ou configurações de controle.

BIBLIOGRAFIA

[1] KRISHNAN, Thadiappan & RAMASWAMI, Bellamkonda. *A Fast Response DC Motor Speed Control System.* IEEE Trans. on Industry Application, Vol.10, No.5, Sept./Oct. 1974.

[2] SEN, Paresh C. & MACDONALD, Murray L. *Thyristorized DC Drives with Regenerative Braking and Speed Reversal.* IEEE Trans. IECI, Vol.25, No.4, November 1978, pp. 347-354.

[3] KRISHNAN T. & RAMASWAMI B. *Speed Control of D.C. Motor using Thyristor Dual Converter.* IEEE Trans. IECI, Vol.23, No.4, November 1976.

[4] BRICKWEDDE, Axel. *Microprocessor-Based Adaptive Speed and Position Control for Electrical Drives.* IEEE Trans. on Industry Application, Vol.21, No.5 Sept./Oct. 1985.

[5] KOFAHL & ISERMANN, R. *A Simple Method to Automatic Tuning of PID-Controllers Based on Parameter Estimation.* American Conference, 1985.

[6] BORGES DA SILVA, Luiz Eduardo & FERREIRA DA SILVA, Valberto. *Proposta de um Controlador Adaptativo para Acionamento de um VAT utilizando a Técnica de Modelo de Referência.* EFEI, 1990.

[7] TADAKUMA, Susumo & TAMURA, Yoshiaki. *Finite-Time Settling Control Based Current Controller for Thyristor Dual Converter.* IEEE Trans. on Industry Application, Vol.23, No.4, Jul./Aug. 1987, pp. 603-609.

[8] NAITOH, Harouc & TADAKUMA, Susumo. *Microprocessor-Based Adjustable-Speed DC Motor Drive using MRAC.* IEEE Trans. on Industry Application, Vol.23, No.2. Mar./Apr. 1987.

[9] BARTHO, F. *Motores Eléctricos Industriales y Dispositivos de Control.* Bilbao, España. Ediciones Urmo, 1968. Traduzido por S. M. García.

[10] DOUVILLE, E. Leo. *Selection and Application of Variable Speed Motor Drive System.* IEEE Trans. on Industry Application, Vol.18, No.6, Nov./Dec. 1982.

[11] MURTY, Y. V. & DUBEY, G. K. *Fault Diagnosis in Three-Phase Thyristor Converters using Microprocessor.* IEEE Trans. on Industry Application, Vol.20, No.6, Nov./Dec. 1984.

[12] DA SILVA, Valberto F. *Apostila de E101 - Eletrônica Industrial II.* Escola Federal de Engenharia de Itajubá. 1984.

[13] MINSKY, Marvin L. & PAPERT, Seymour A. *Perceptrons: An Introduction to Computational Geometry.* The MIT Press, 1989, Third Printing, 1988.

[14] LIPPMAN, Richard P. *An Introduction to Computing with Neural Nets.* IEEE ASSP Magazine, April 1987.

[15] RANGWALA, Sabbir S. & DORNFELD, David A. *Learning and Optimization of Machining Operation using Computing Abilities of Neural Networks.* IEEE Trans. on Systems, Man and Cybernetics. Vol. 19, No.2, Mar./Apr. 1989.

[16] RUMELHART, David E. & MCCLELLAND, James L. Parallel *Distributed Processing - Vol.1: Foundations.* The MIT Press, 1969. Third Printing, 1988.

[17] PAO, Yo-Han. *Adaptive Pattern Recognition and Neural Networks.* Addison-Wesley Publishing, Inc. U.S.A. 1989.

[18] KRUSHKE, John K. & MOVELLAN, Javier R. *Benefits of Gains: Speeded Learning and Minimal Hidden Layers in Back-Propagation Nets.* IEEE Trans. on Systems, Man and Cybernetics. Vol.21, No.1, Jan./Feb. 1991.

[19] AZIMI-SADJADI, Mahmood R. & LIOU, Ren-Jean. *Fast Learning Process of Multilayer Neural Networks using Recursive Least Squares Method.* IEEE Trans. on Signal Processing, Vol. 40, No.2, Feb. 1992.

[20] HUANG, Shih-Chi & HUANG, Yih-Fang. *Learning Algorithm for Perceptron using Back-Propagation with Selective Updates.* IEEE Control Systems Magazine, April 1990.

[21] GORI, M. & TESI, A. *On the Problem of Local Minima in Back-Propagation.* IEEE Trans. on Pattern Analysis and Machine Intelligent, Vol. 14, No.1, Jan. 1992, pp. 76-86.

[22] KHOTANZAD, Alireza & LU, Jing-Her. *Classification of Invariant Images Representation using Neural Networks.* IEEE Trans. ASSP, Vol.38, No.6, June 1990.

[23] WIDROW, Bernard & WINTER, Rodney. *Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition.* IEEE Computer Magazine, March 1988.

[24] ANDERSEN, Kristian & RAMASWAMY, Kumar. *Artificial Neural Networks Applied to Arc Welding Process Modeling and Control.* IEEE Trans. on IA, Vol.26, No.5, Sept./Oct. 1990.

[25] BHAT, Naveen V. & MINDERMAN, Peter A. *Modeling Chemical Process Systems via Neural Computation.* IEEE Control System Magazine, April 1990, pp. 24-30.

[26] ALVES DA SILVA, Alexandre P. ,QUINTANA, Victor H. & PANG, G. K. h. *Neural Networks for Topology Determination of Power Systems.* Proc. of the First International Forum on Applications of Neural Networks to Power Systems. Seattle, Washington. July 1991.

[27] LAMBERT TORRES, G. & MUKHEDKAR, D. *Application of Neural Networks for Load Forecasting.* Sixth National Power Systems Conference, VI NPSC, Bombay, India, Jun. 4-7, 1990.

[28] SORSA, Timo & KOIVO, Keikki N. *Neural Networks in Process Fault Diagnosis.* IEEE Trans. on Systems, Man and Cybernetics, Vol.21, No.4, Jul./Aug. 1991.

[29] SALAM, Fathi & WANG, Yimen. *A Real Time Experiment using a 50-Neuron CMOS Analog Silicon Chip with On-Chip Digital Learning.* IEEE Trans. on Neural Networks, Vol.2, No.4, July 1991, pp. 461-464.

[30] VIDAL, Jacques J. *Implementing Neural Nets with Programable Logic.* IEEE Trans. ASSP, Vol.36, No.7, July 1988.

[31] HWANG, Jeng-Neng & VLONTZOS, John A. *A Systolic Neural Network Architecture for Hidden Markov Models.* IEEE Trans. ASSP, Vol.37, No.12, December 1989.

[32] FORREST, B. M. & WALLACE, D. J. *Implementing Neural Networks Models on Parallel Computers.* The Computer Journal, Vol.30, No.5, 1987, pp. 413-419.

[33] HANDELMAN, David A. & LANE, Stephen H. *Integrating Neural Networks and Knowledge-Based Systems for Intelligent Robotic Control.* IEEE Control Systems Magazine, April 1990.

[34] NGUYEN, Derrick H. & WIDROW, Bernard. *Neural Networks for Self-Learning Control Systems.* IEEE Control Systems Magazine, April 1990.

[35] NARENDRA, Kumpaty & PARTHESARATHY, Kannan. *Identification and Control of Dynamical Systems Using Neural Networks.* IEEE Trans. on Neural Networks, Vol.1, No.1, March 1990.

[36] CHEN, Fu-Chuang. *Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control.* IEEE Control Systems Magazine, April 1990.

[37] KRAFT, L. Gordon & CAMPAGNA, David P. *A Comparison Between CMAC Neural Networks Control and Two Traditional Adaptive Control Systems*. IEEE Control Systems Magazine, April 1990.

[38] HUNT, K. J. & SBARBARO, D. *Neural Networks for Nonlinear Internal Model Control*. IEE Proceedings-D, Vol.138, No.5, September 1991.

[39] NAIDU, Sinassamy R. & ZAFIRIOUS, Evanghelos. *Use of Neural Networks for Sensor Failure Detection in a Control System*. IEEE Control Systems Magazine, April 1990.

[40] MILLER III, W. Thomas. *Real-Time Applications of Neural Networks for Sensor-Based Control of Robots with Vision*. IEEE Trans. on Systems, Man and Cybernetics., Vol.19, No.4 Jul./Aug. 1989.

[41] SAERENS, M. & SOQUET, A. *Neural Controller Based on Back-Propagation Algorithm*. IEE Proceedings-F, Vol.138, No.1, February 1991.

[42] PSALTIS, Demetri & SIDERIS, Athanasios. *A Multilayered Neural Network Controller*. IEEE Control Systems Magazine, April 1988.

[43] BAILEY, David, & THOMPSON, Donna. *How to Develop Neural-Network Applications*. AI Expert Magazine, Vol.5, No.6, June 1990, pp. 38-47.

[44] BUR, David J. *Experiments on Neural Nets Recognition of Spoken and Writen Text*. IEEE Trans. ASSP, Vol.36, No.7, July 1988.

[45] WANG, J. & MALAKOOTI, B. *On Training of Artificial Neural Networks*. IJCNN International Joint Conference on Neural Networks, Washington D. C., Vol.II, July 1989.

[46] CAUDILL, Maureen. *Neural Networks Tips and Techniques.* AI Expert Magazine, Vol.6, No.1, January 1991, pp.56-61.

[47] HOUPIS, Constantine & LAMONT, Gary. *Digital Control Systems: Theory, Hardware, Software.* McGraw-Hill Book Company, 2nd Printing 1987, Singapure.

[48] KUO, B. C. *Discrete-Data Control Systems.* Prentice-Hall, Inc. Printed in U.S.A, 1970.

[49] ISSERMAN, Rolf. *Digital Control Systems.* Springer-Verlag Berlin Heidelberg New York, 1981.

[50] LANDAU, Yoan D. *Adaptive Control (Control System Theory; v.8).* Marcel Dekker, Inc. New York, 1979.

[51] LAMBERT TORRES, G. & MUKHEDKAR, D. *Measurement Validation by Neural Networks.* Proceedings of IEEE Mexicon´89 Meeting, Mexico City-Mexico, Sept. 1989.

[52] CASTILLO S., Ernesto A., LAMBERT TORRES, Germano & BORGES DA SILVA, Luiz Eduardo. *Identificação e Controle de um Acionamento de Motor CC usando Redes Neurais.* 9º Congresso Brasileiro de Automática (CBA), Vitória-ES. Setembro 1992.

[53] CASTILLO S., Ernesto A., LAMBERT TORRES, Germano, BORGES DA SILVA, Luiz Eduardo & Do, X. D. *An Adaptive Neural Net Controller for DC Motor Drive.* 2nd International Forum on Applications of Neural Networks to Power Systems. Yokohama, Japan. April 1993. (Em aprovação).

[54] CASTILLO S., Ernesto A., LAMBERT TORRES, Germano, BORGES DA DA SILVA, Luiz Eduardo & Do, X. D. *Identification and Control of a DC Motor using Neural Networks.* 12th World Congress International Federation of Automation Control. Sydney, Australia. July 1993. (Em aprovação).

APÊNDICE A

Algumas diretivas da Transformada de Laplace

Diretiva 1: Homogeneidade.

$$\mathcal{L}[af(t)] = a\mathcal{L}[f(t)] = aF(s)$$

Diretiva 2: Derivada real.

$$\mathcal{L}[Df(t)] = sF(s) - f(0^+)$$

Diretiva 3: Integração real.

$$D^{-1}f(t) = \int_0^t f(t)dt + D^{-1}f(0^+)$$

$$\mathcal{L}[D^{-1}f(t)] = \frac{F(s)}{s} + \frac{D^{-1}f(0^+)}{s}$$

Tabela de Transformadas Z

Função no tempo	Trans. de Laplace	Transformada Z
$e(kT)\delta[t-(k+1)T]$	$e^{-1Ts}.E(s)$	$z^{-1}.E(z)$
$u_{-1}(t)$ / $\delta(t - KT)$	$\frac{1}{s}$	$\frac{z}{z - 1}$
e^{-at}	$\frac{1}{s + a}$	$\frac{z}{z - e^{-aT}}$

Algumas propriedades da Transformada Z

Propriedade 1: Adição e Substração.

$$Z\ [\ e_a^*(t) + e_b^*(t)\ ] = E_a(z) + E_b(z)$$

Propriedade 2: Multiplicação por uma constante.

$$Z\ [\ Ae^*(t)\ ] = A \cdot Z\ [\ e^*(t)\ ] = A \cdot E(z)$$

Propriedade 3: Deslocamento.

$$Z\ [\ e^*(t - pT)\ ] = z^{-p} \cdot E(z)$$

APÊNDICE B

NEUWORK — DESENVOLVIMENTO DE APLICAÇÕES UTILIZANDO REDES NEURONAIS

NEUWORK é um pacote de ferramentas com interface totalmente amigável, orientado para o desenvolvimento de aplicações de redes neuronais na área de controle.

O pacote é constituído por três programas: NEUWORK I, NEUWORK II e NEUCONTROL, para computadores IBM PC ou compatíveis, sob ambiente DOS 2.0 ou versão superior. Necessita-se um mínimo de 300 Kbytes no disco para os arquivos executáveis e de dados. O pacote utiliza alocação dinâmica de memória, o que possibilita um melhor aproveitamento da memória RAM.

O pacote cria e manipula arquivos com extensão:

- .neu -----> estrutura de rede.
- .par -----> conjunto de pesos.
- .dat -----> conjunto de padrões para treinamento "off-line".
- .eqd -----> estrutura e parâmetros de equações as diferenças.

NEUWORK I

NEUWORK I é um programa destinado ao treinamento tipo "off-line" de redes neuronais do tipo multicamadas, permitindo, entre outras funções, a criação, treinamento e teste das redes. O treinamento é realizado através do algoritmo "back-propagation" utilizando várias técnicas de treinamento rápido.

A tela principal do programa, figura B.1, mostra a estrutura, em termos de camadas e número de neurônios, da rede.

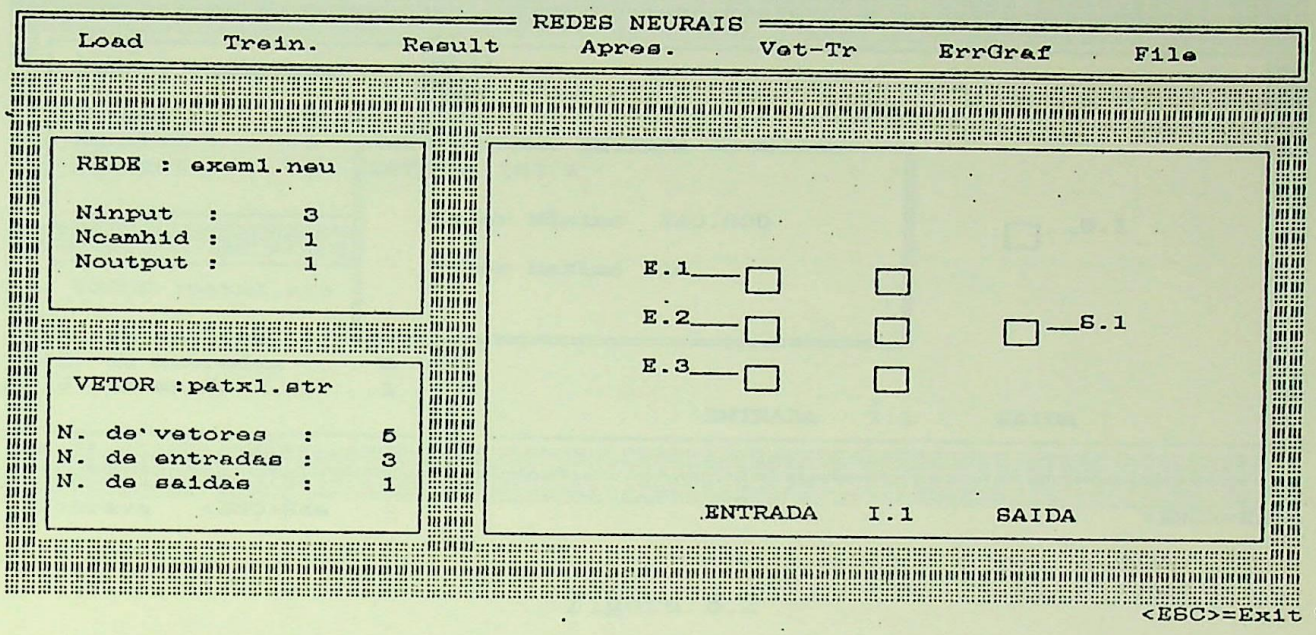


Figura B.1

Através da opção TREIN. do menu principal podem ser ajustadas as faixas de normalização (figura B.2) e os parâmetros do treinamento (figura B.3), entre eles, o valor das taxas de treinamento, o erro global mínimo desejado e o número máximo de iterações. Estão disponíveis para o treinamento, duas opções de aceleração do algoritmo, através de taxas de treinamento variável e através de ganhos auxiliares.

O conjunto de padrões de treinamento pode ser selecionado para treinamento, construído e visualizado através da opção VET-TR.

O treinamento pode ser acompanhado através das saídas da rede, das saídas desejadas e dos erros parciais e globais, atualizados em intervalos de tempo definidos pelo usuário (figura

B.4).

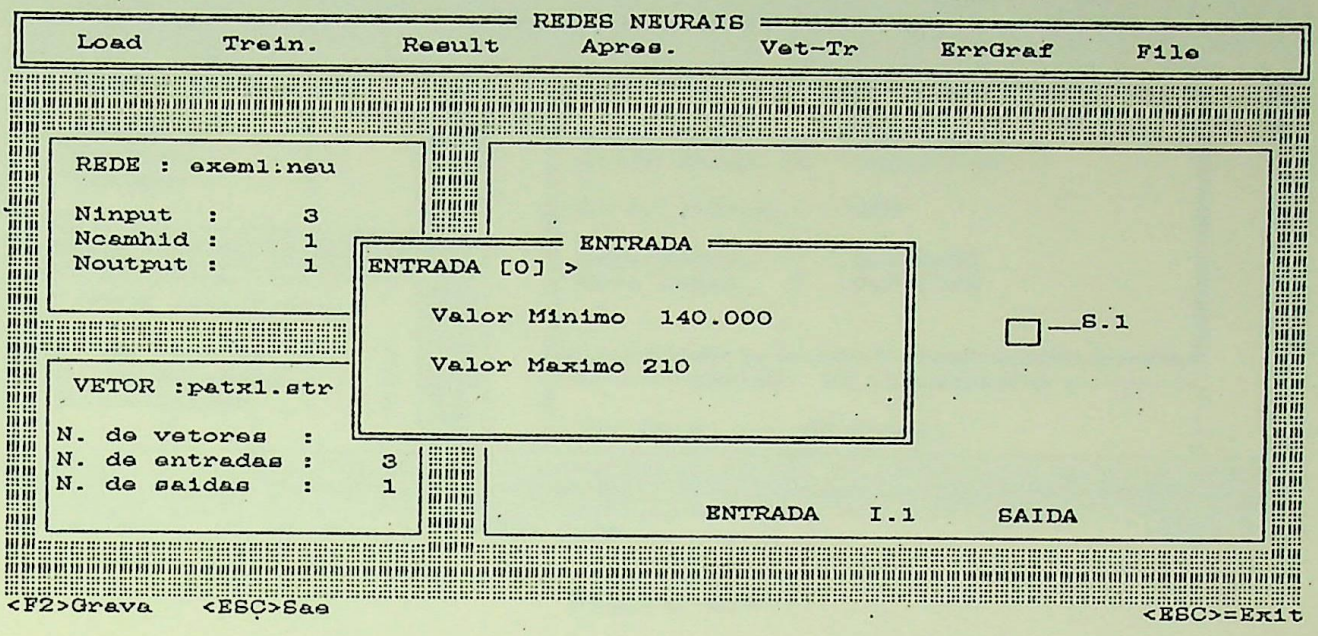


Figura B.2

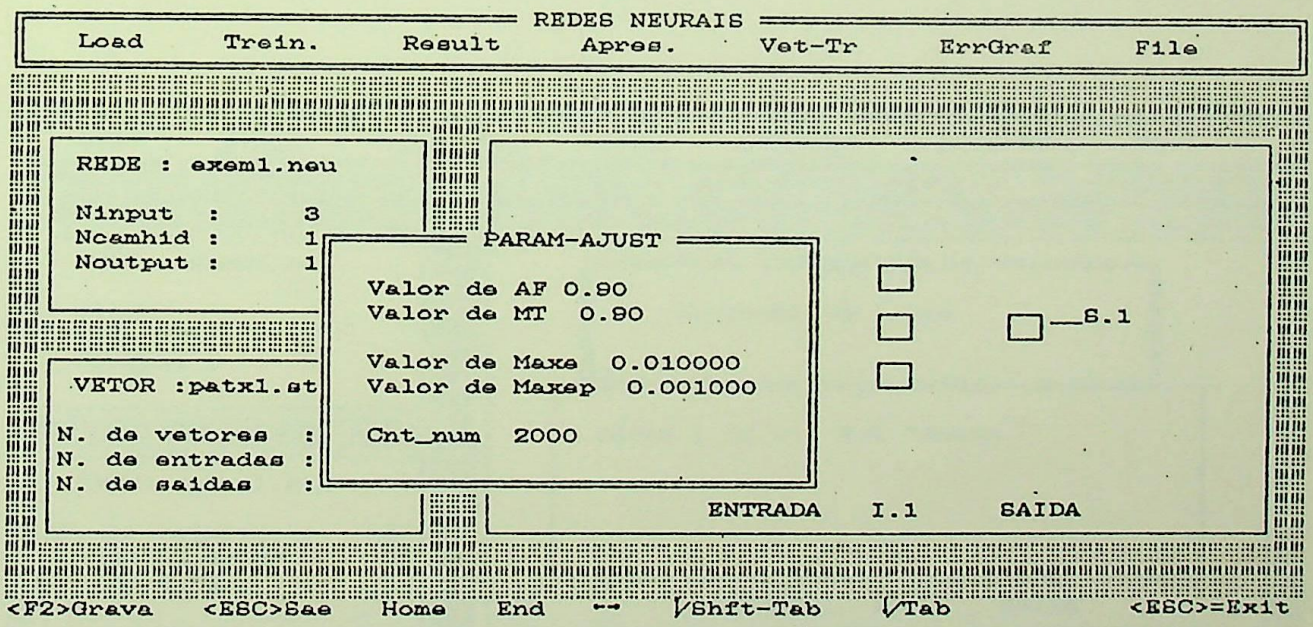


Figura B.3

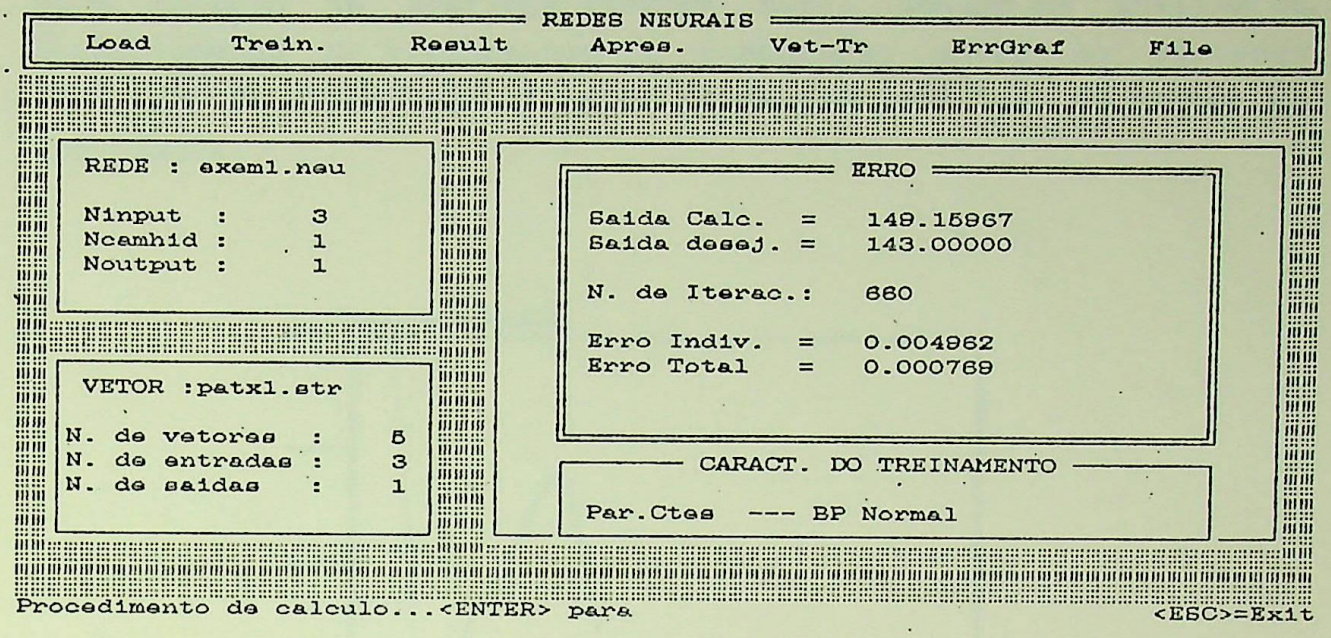


Figura B.4

Após o treinamento, a rede pode ser testada utilizando a opção RESULTADO com entradas através de janelas (figura B.5).

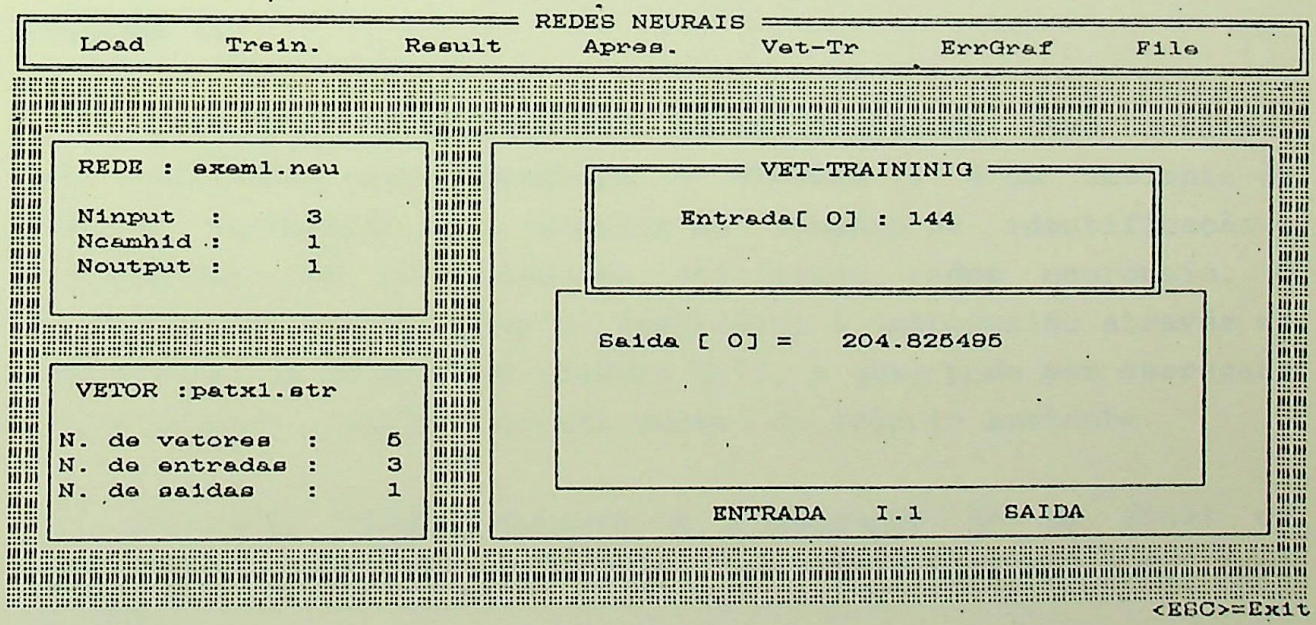


Figura B.5

O programa permite ainda a criação de arquivos de erros para geração de gráficos (figura B.6), usados na análise do comportamento do treinamento e comparação entre as diversas técnicas de acelaração. A saída pode ser tanto na tela como na impressora.

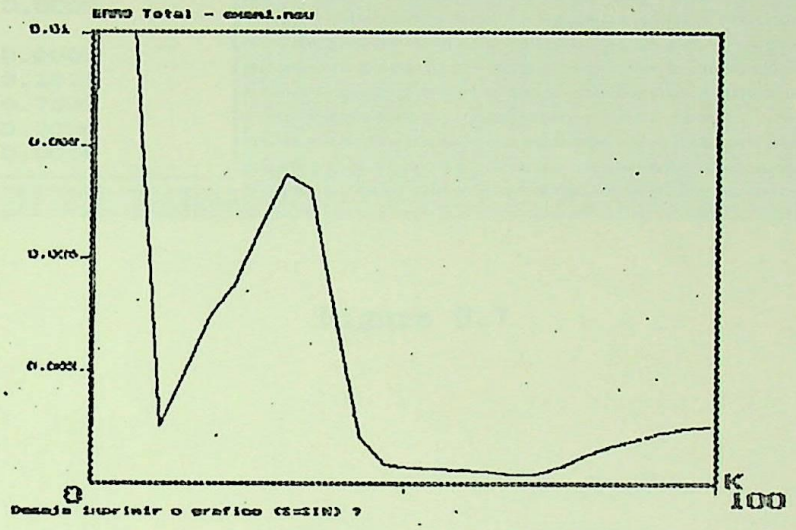


Figura B.6

NEUWORK II

Simulações de treinamento em aplicações do tipo "on-line" são realizadas neste programa. O NEUWORK II é um ambiente de simples utilização para simular as funções de identificação e treinamento de controladores utilizando redes neuronais. O sistema a ser identificado e controlado é introduzido através de sua equação as diferenças (figura B.7), a qual pode ser carregada de um arquivo (.eqd) ou criada dentro do próprio ambiente.

A opção REFER. permite a construção de um sinal de referência constituído por degraus ou rampas ou funções do tipo senoidal.

O programa possue uma interface gráfica para cada um das opções de treinamento. A identificação (figura B.8) permite a

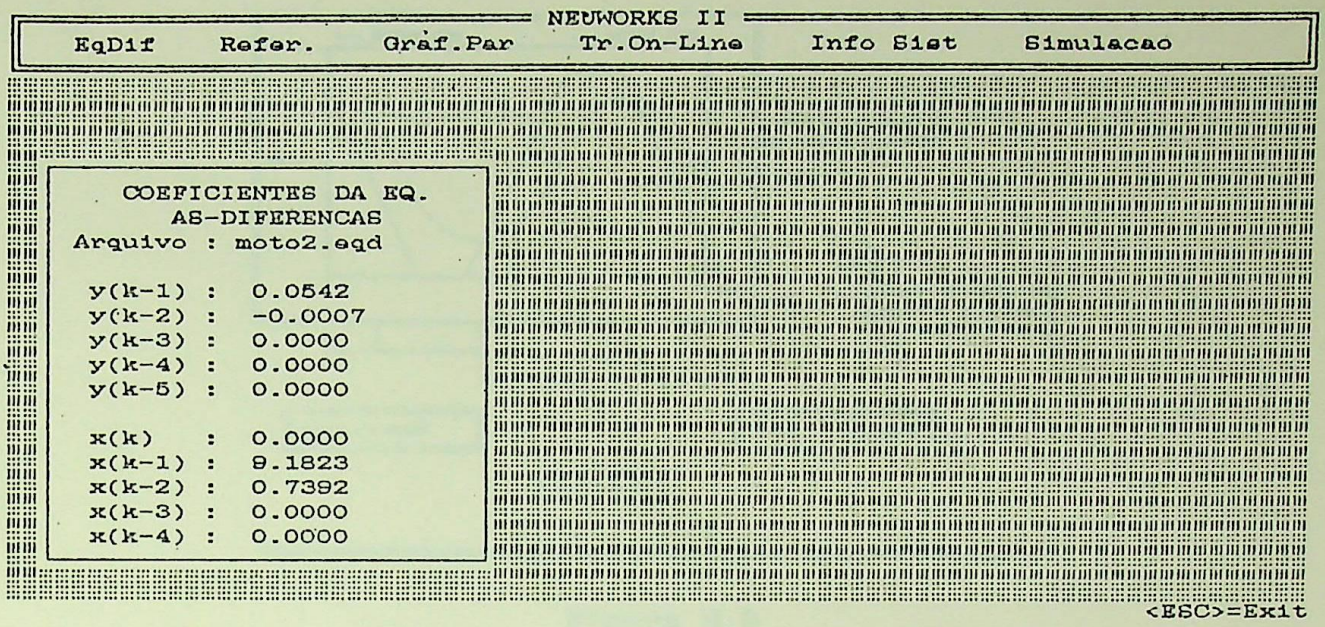


Figura B.7

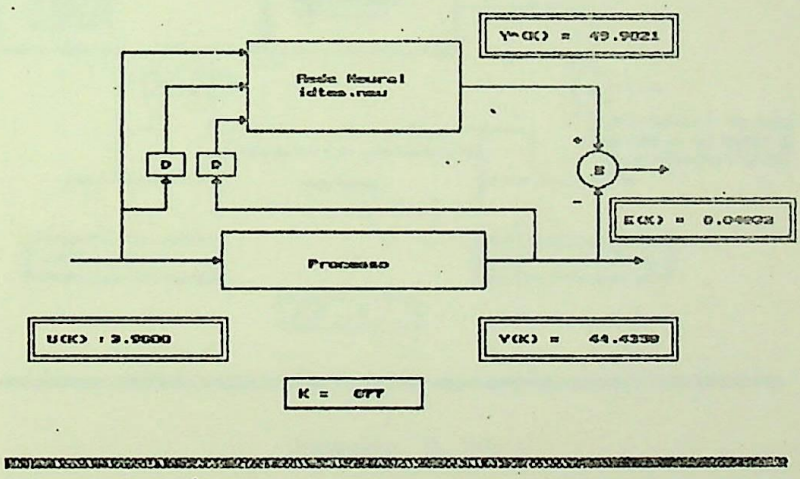


Figura B.8

utilização de várias configurações para a rede, assim como o acompanhamento do procedimento, não só através dos valores dos sinais mais importantes a cada iteração, mas também, através de gráficos (figura B.9) e da variação dos pesos (figura B.10). O treinamento do controlador (figura B.11) possue os mesmos recursos mostrados no procedimento de identificação.

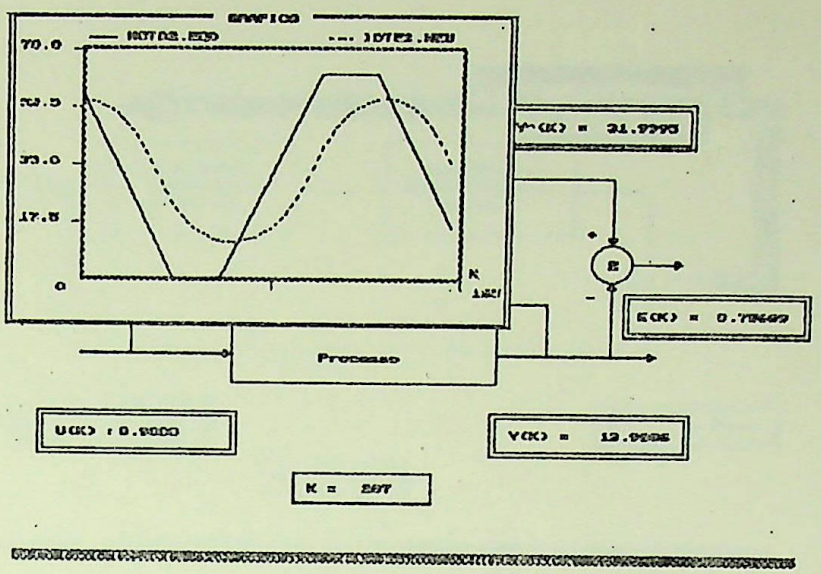


Figura B.9

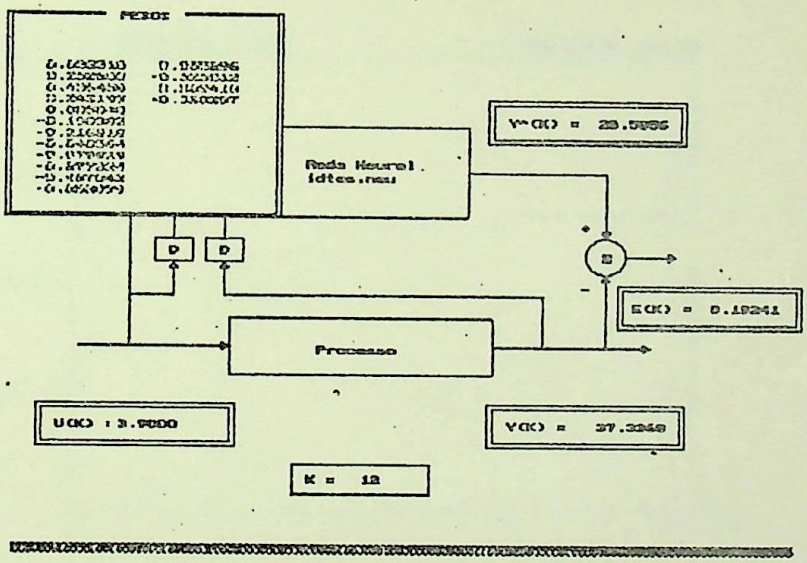


Figura B.10

Os resultados podem ser conferidos através de simulação em gráficos tanto na tela (figura B.12) como na impressora, identificando os sinais envolvidos na parte superior do gráfico.

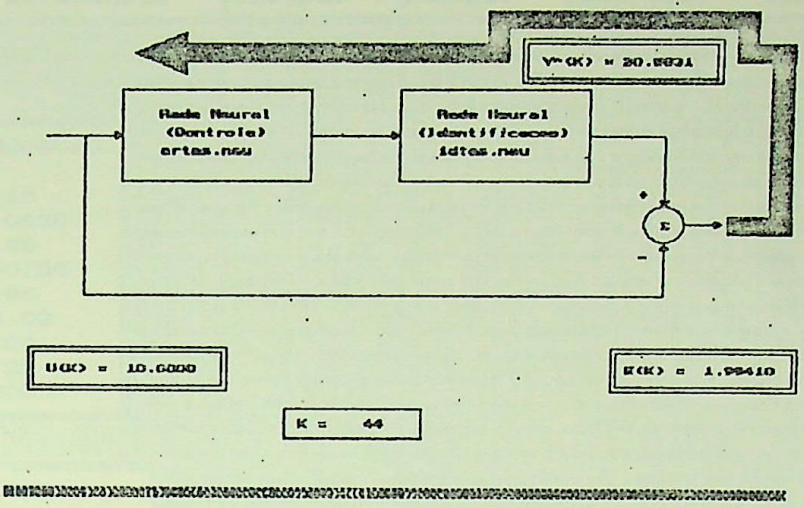


Figura B.11

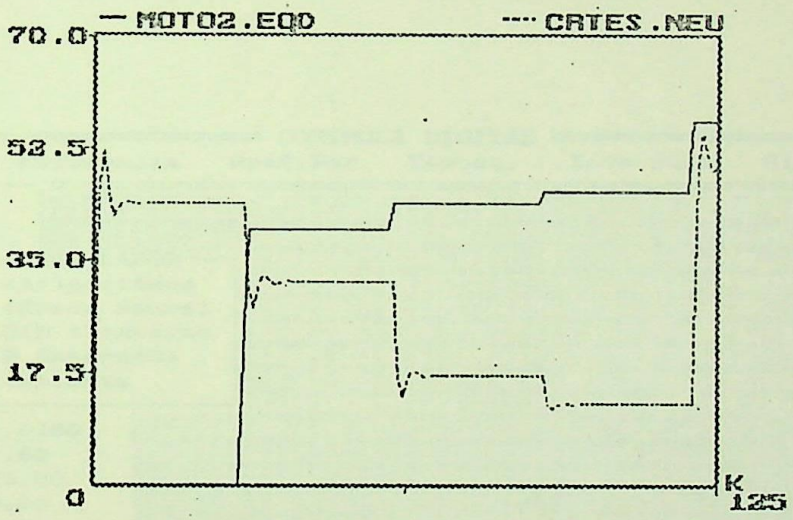


Figura B.12

NEUCONTROL

Finalmente, o programa NEUCONTROL é um simulador da malha de controle de velocidade de um acionamento de motor CC, cujos parâmetros podem ser introduzidos e modificados fácilmente (figura B.13).

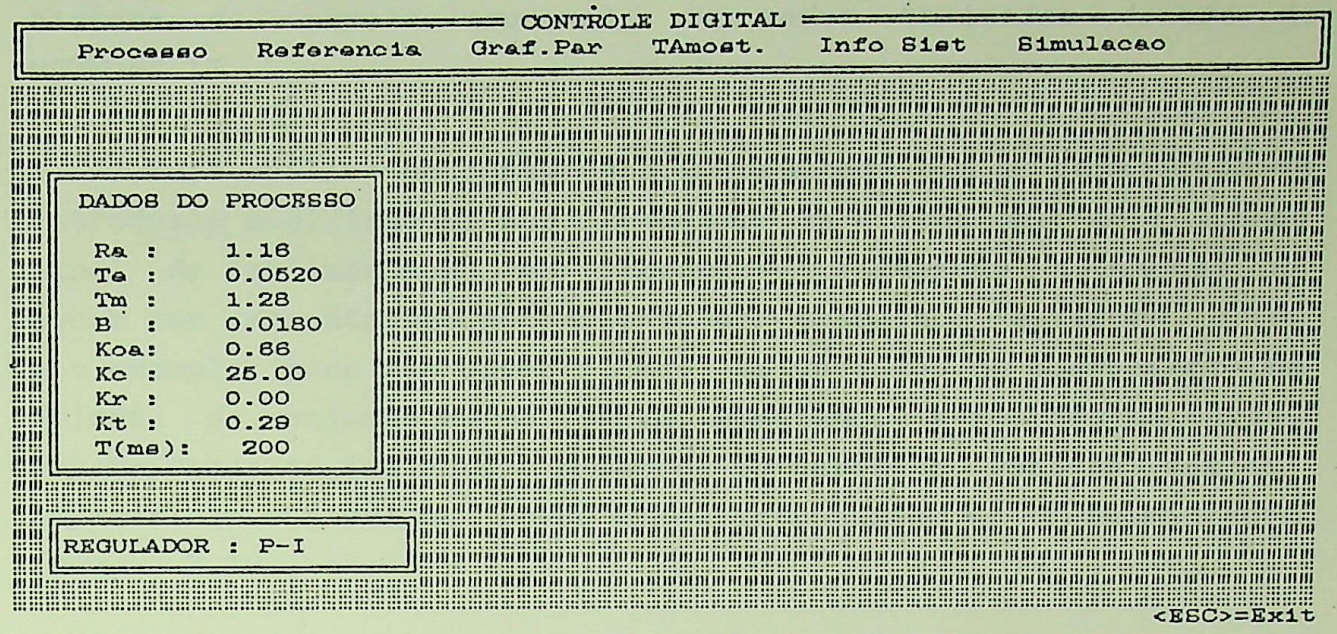


Figura B.13

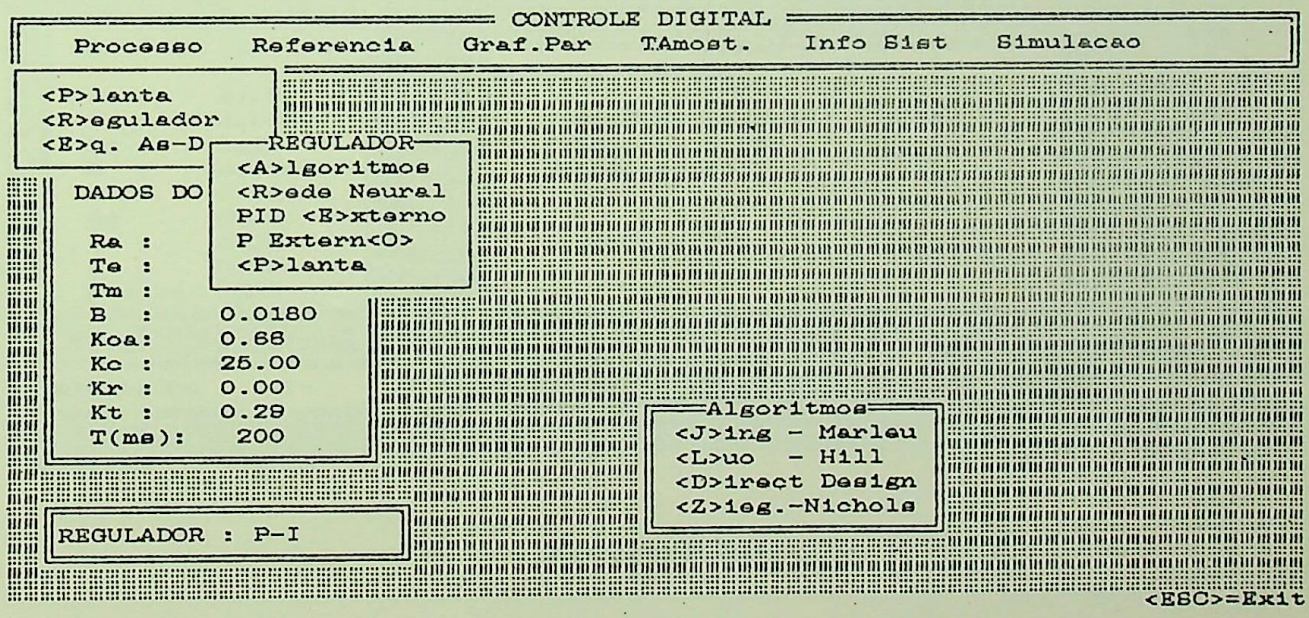


Figura B.14

O programa inclui a escolha do tipo de controlador (figura B.14) entre vários tipos conhecidos, entre técnicas de resposta rápida "deadbeat response" ou inclusive a utilização de um

sistema de controle. neuronal, com redes treinadas dentro do NEUWORK II.

Para análise de desempenho, o programa tem a capacidade de introduzir modificações nos parâmetros do acionamento ou ruído no sinal de realimentação do transdutor, simulando situações que podem ser encontradas no ambiente de trabalho. Na figura B.15, por exemplo, pode ser especificada uma variação no coeficiente de fricção do conjunto motor carga, simulando variações na carga, determinando o instante e o tempo de variação. Pode-se também, especificar o nível de ruído a ser introduzido na realimentação (figura B.16).

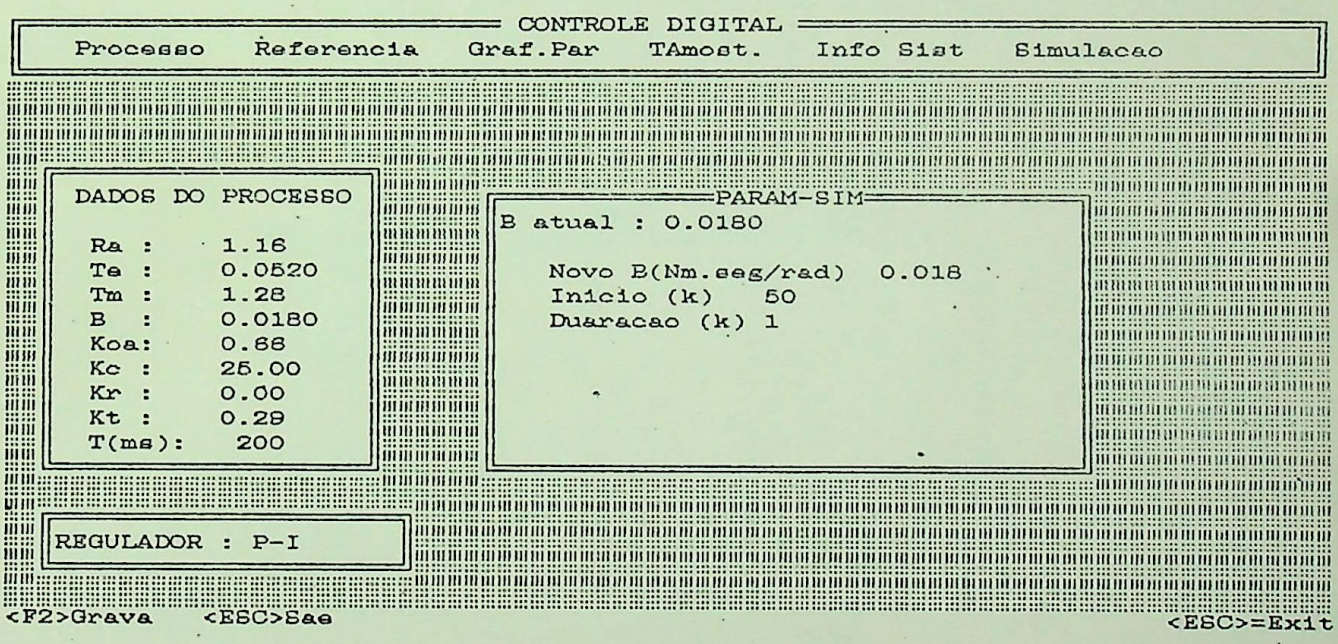


Figura B.15

O programa possue também, saídas gráficas das simulações, tanto na tela como na impressora, da resposta da malha a uma referência especificada (figura B.17).

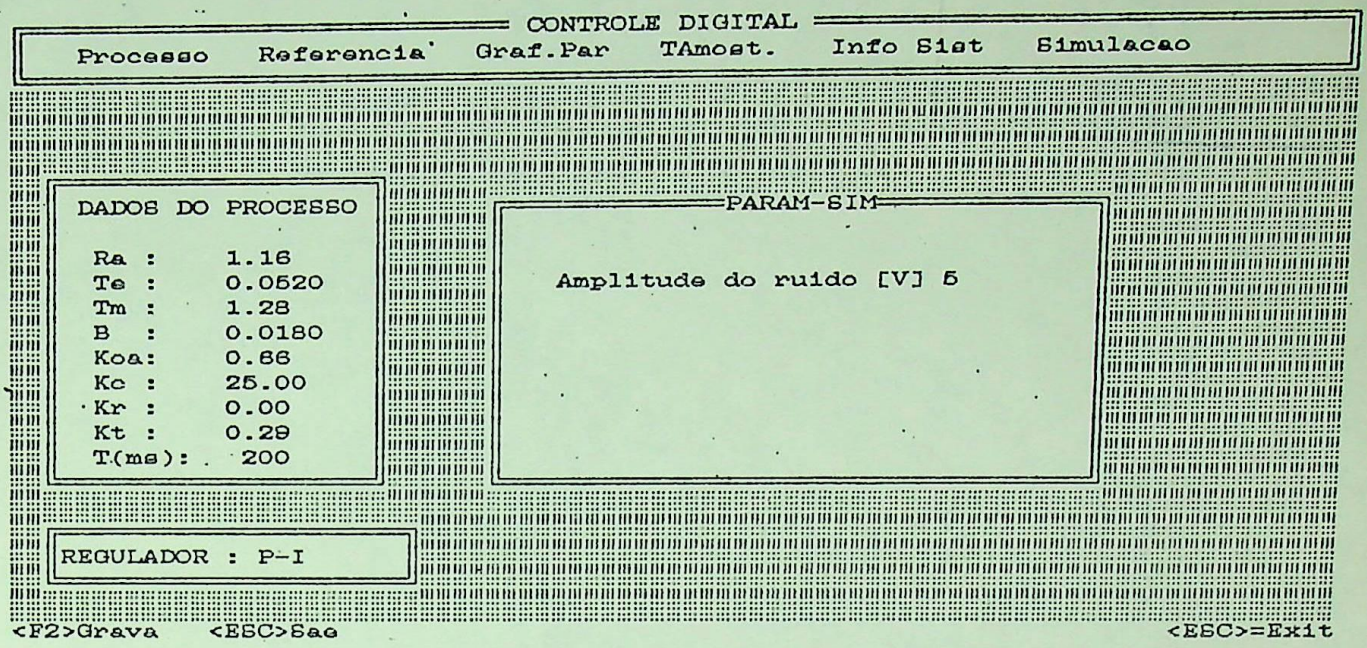


Figura B.16

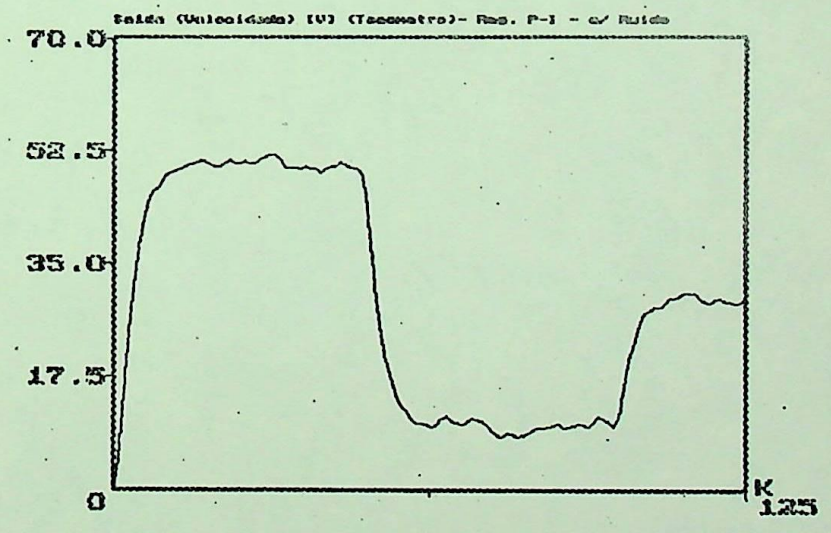


Figura B.17

DATA 03/03/93
PROC.
FED. PPG
LIV.
CzS DOACAO

EFEI - BIBLIOTECA MAUÁ
8200694
NÃO DANIFIQUE ESTA ETIQUETA