ESTIMAÇÃO DE PARÂMETROS DE UM MOTOR CC POR MEIO DE REDES NEURAIS ARTIFICIAIS (RNA)

Cleber de Medeiros Navarro[1]
e-mail: cleber.navarro@ig.com.br

Adelson Siqueira Carvalho[2]
e-mail: adelsonsc@yahoo.com.br

Geraldo Galdino de Paula Junior[1]
e-mail: galdino@uenf.br

[1]Universidade Estadual do Norte Fluminense - UENF
www.uenf.br
[2]Instituto Federal de Educação, Ciência e Tecnologia Fluminense - IFF
www.iff.edu.br

RESUMO

Este trabalho apresenta a metodologia e resultados da estimação de parâmetros para um motor CC do Kit Feedback® – Servo Fundamentals Trainer. Trata-se de um kit didático que possibilita colocar em prática várias técnicas de controle, como por exemplo Controle Clássico, Moderno e Avançado. Simulando esse motor CC como um sistema dinâmico real (atuador de satélite, bobinadeiras, guindaste e etc), mas sem perder significativamente suas características reais. Neste artigo propõe-se a técnica de estimação de parâmetros por RNA (Redes Neurais Artificiais) do tipo *feedforward* multicamadas com aprendizado supervisionado pelo algoritmo de treinamento *backpropagation*. Para validar o projeto da RNA, o índice de desempenho MSE (Mean Square Error) foi analisado com a intenção de verificar a menor diferença entre o sistema e o modelo estimado. Esses resultados foram compilados na forma gráfica e numérica, onde é confrontado o desempenho do modelo em relação ao sistema.

PALAVRAS-CHAVE. RNA. estimação de parâmetros. motor CC. MH – Metaheurística

ABSTRACT

This work presents the metodology and results of parameters estimation for a DC motor from the Feedback® - Servo Fundamentals Trainer kit. This is about an didactic kit that allows to put into practice several techniques of control, as for example Classic control, Modern and Advanced. Simulating this DC motor as a true dinamic system (satellite handler, winding frames, cranes, etc), but not loosing it's actual meaningful characteristics. This article proposes the technique of parameter estimation by ANN (Artificial Neural Network) of *feedfoward* multi-layers type with supervised learning by *backpropagation* training algorithm. To validate the ANN project, the performance index MSE (Mean Square Error) was analyzed to verify the smaller difference between the system and the pattern estimated. Those results were compiled in graphic and numeric way, where is confronted the performance respecting the system.

KEYWORDS. ANN. parameter estimation. DC motor. MH – Metaheurística

1. Introdução

Os motores CC (motores de corrente contínua) têm grande utilidade no desenvolvimento de servossistemas (LEONHARD, 2001). Isso devido à sua robustez, alto torque na partida e em baixas rotações, facilidade de controlar a sua velocidade, entre outras vantagens (NAVARRO, 2008). Com sua grande aplicação no meio industrial, como em ferramentas de avanço, bobinadeiras, máquinas têxteis, guinchos e guindastes, exaustores entres várias outras (WEG, 2004). Diversos métodos de estimação e controle desses motores são estudados e desenvolvidos, com a finalidade de otimização do sistema ao qual ele está aplicado.

Sabe-se que a estratégia de controle de processos tem a finalidade de otimizar um sistema baseada em especificações pré determinada no instante de um projeto. Ou seja, ele faz com que o processo trabalhe seguindo um objetivo desejado (set point). Para maior eficácia desses sistemas de controle é preciso um estudo minucioso das variáveis envolvidas e dos equipamentos utilizados para poder modelar seus comportamentos. Esses estudos resultam em modelos matemáticos, para que a partir desse ponto seja inicializado o projeto do controlador.

Um modelo mal estimado interfere intrinsecamente em baixa eficiência no sistema. Existem vários métodos de estimação de parâmetros de sistemas dinâmicos, como é o caso do motor CC. Dentre esses métodos, os estatísticos são amplamente usados, um exemplo é o *método dos mínimos quadrados*, mais informações sobre esse método de estimação em motores CC ver Basílio e Moreira (2001).

No ramo da pesquisa operacional, esses modelos matemáticos estão ligados no auxílio à tomada de decisão. Suas características definem o grau de melhoria no processo de otimização desses sistemas na prática. Por isso, deve-se ter atenção desdobrada no processo de modelagem, pois qualquer estudo que depende dos dados estimados ficaria com baixo rendimento.

A proposta nesse artigo é utilizar o método de estimação baseado em inteligência artificial, mais precisamente Redes Neurais Artificiais (RNA) para a identificação do modelo de um motor CC. A RNA utiliza como entrada para o treinamento os sinais da resposta de velocidade do motor CC coletados em laboratório, para diversos valores de tensão aplicados em sua armadura. Com a intenção de aproximar ao máximo a resposta do modelo estimado com o sistema real. Reduzindo assim o MSE a um valor mais próximo de zero possível.

O MSE representa a diferença entre os valores estimados y pelo os valores do sistema x, elevado ao quadrado dividido pelo número de amostras n. Ou seja:

$$MSE = \frac{\sum_{i=1}^{n}(x-y)^2}{n} \tag{1}$$

O restante da estrutura deste artigo será apresentado na seguinte maneira: na seção 2 a fundamentação teórica será abordada, fazendo um breve comentário sobre a estrutura das RNA's enfatizando a sua aplicação em modelagem de sistemas dinâmicos, em especial o motor CC. Apresentando ainda o Kit Feedback® – Servo Fundamentals Trainer no qual se encontra o servossistema. Na seção 3 um breve estado da arte será comentado, visando observar a evolução dos estudos referentes à modelagem de sistemas dinâmicos utilizando RNA's. Na seção 4 é apresentada a metodologia e os materiais utilizados para a estimação dos parâmetros do motor CC. Já na seção 5 estão os resultados obtidos a partir das simulações, onde são confrontadas as respostas do modelo dinâmico do motor CC juntamente com as respostas do modelo estimado. Todos os resultados são apresentados na forma de gráficos e ainda na forma numérica por meio do índice de desempenho MSE. Por último, na seção 6, estão as considerações finais e as perspectivas apresentadas para um futuro desenvolvimento.

2. Fundamentação Teórica

Uma RNA (Redes Neurais Artificiais) é também conhecida como um sistema conexionista. Que segundo Haykin (2001) são modelos computacionais inspirados na estrutura e funcionamento do cérebro humano e caracterizam-se pela reunião de uma grande quantidade de células (elementos ou unidades) de processamento interligadas por um grande número de conexões (links), que processam a informação de forma paralela. E ainda, esse tipo de processamento é chamado de Processamento Paralelo Distribuído (PPD). Segundo Corrêa (1995) a principal finalidade das RNA's é explorar e reproduzir tarefas humanas naturais em processamento de informações. Além disso, elas também são usadas para solução de problemas de otimização combinatória, modelagem de sistemas, controle e aproximação de funções. De maneira geral, uma RNA tenta "simular" um sistema nervoso humano em forma de modelos matemáticos e programas de computador. Segundo Haykin (2001), ela se assemelha com o cérebro em duas formas:

- O conhecimento é adquirido pela rede a partir de seu ambiente por meio de um processo de aprendizagem;
- Forças de conexão entre os neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

O processo de aprendizagem é realizado por meio de treinamento. Podendo esse ser caracterizado como aprendizado supervisionado ou não. O nome dado ao procedimento de aprendizagem é chamado de algoritmo de treinamento e sua função é modificar os pesos sinápticos das redes de forma ordenada para alcançar um objetivo de projeto desejado.

Segundo Freeman e Skapura (1991) o que define o tipo de aprendizado supervisionado é que nesse modelo são apresentadas à rede padrões de entrada e seus correspondentes padrões de saída. Assim, a rede ajusta os pesos das conexões seguindo uma determinada lei de aprendizagem predeterminada. Isso é feito até que o desvio entre os padrões de saída gerados pela rede atinja um valor mínimo tolerado. Ainda segundo Freeman e Skapura (1991), existem dezenas de leis de aprendizagem supervisionada como a lei do modelo *perceptron*, a *backpropagation* etc.

Já no aprendizado não-supervisionado, são apresentados os conjuntos de dados a rede, ela analisa, determina algumas propriedades desses conjuntos e aprende a refletir estas propriedades na sua saída. Ela agrupa os conjuntos de dados em classes, de acordo com padrões, regularidades e correlações. As propriedades que a rede vai "aprender" sobre os dados pode variar em função do tipo de arquitetura utilizada e da lei de aprendizagem. Alguns métodos de aprendizado não-supervisionado são Mapa Auto-Organizavél de Kohonen, Redes de Hopfield e Memória Associativa Bidirecional (FREEMAN & SKAPURA, 1991).

Segundo Medeiros (1999) *apud* Caudill (1991) para o desenvolvimento de um modelo ou projeto baseado em redes neurais artificiais são necessárias diversas etapas como: definição do problema; coleta dos dados de treinamento e de teste; pré e pós-processamento dos dados; projeto da estrutura da rede; treinamento; teste e validação.

A questão a ser observada na elaboração do projeto de uma RNA é determinar o número de elementos de processamento da camada oculta, assim como o número de camadas ocultas. Porém, segundo Beale e Jackson (1992) estudos demonstraram que uma única camada oculta é suficiente para representar qualquer função ou para resolver a maioria dos problemas de generalização. Mas, o que varia é o número de elementos de processamento que esta camada deve conter. Então, antes que defina exatamente o número de neurônios a ser usado o ideal é que se tenha um conhecimento prévio do sistema a ser identificado, assim como o conjunto de dados que ele oferece para o treinamento. Isso interfere diretamente no desempenho e na característica da rede e por esse motivo deve ser observado.

Devido a recursos tais como torque elevado, possibilidade de controle de velocidade sobre uma ampla faixa de valores, portabilidade, característica velocidade e torque bem comportada e adaptabilidade a vários tipos de métodos de controle, os motores CC são largamente usados em numerosas aplicações de controle, incluindo manipuladores robóticos, mecanismos de transporte de fitas, acionadores de disco, máquinas-ferramentas e atuadores de servoválvulas. Por isso, a identificação de um modelo adequado para um possível desenvolvimento de um sistema de controle deve ser obtida.

Um motor CC é um dispositivo de potência que entrega energia mecânica a uma carga. Ele converte energia elétrica de corrente contínua (CC) em energia mecânica rotativa. Uma fração importante do torque gerado no rotor do motor está disponível para acionar uma carga externa. Seu funcionamento é feito através da variação de intensidade de tensão inserida na armadura do mesmo. Gerando assim uma velocidade proporcional a tensão aplicada em seus terminais. A faixa de operação do motor CC utilizado neste trabalho é de –10V a +10V. Para maiores detalhes sobre o funcionamento de motores CC ver (DORF & BISHOP, 2004).

O Kit Feedback® – Servo Fundamentals Trainer composto por uma unidade mecânica (UM-33-100), uma unidade analógica (UA-33-110), uma unidade digital (UD-33-120) e ainda o software Discovery 33-910 é um ótimo equipamento para desenvolvimento de pesquisas e práticas de laboratórios, pois o mesmo simula de maneira aproximada e eficiente um sistema real.

Como elementos para a pesquisa serão utilizados um microcomputador (PC), para que sejam configuradas todas as estratégias desenvolvidas, além da aquisição dos dados. Na UM-33-100 será usado o atuador (motor CC), o sensor de velocidade (tacogerador) que se encontra acoplado ao eixo do motor CC e ainda a eletrônica de suporte da unidade. Já na UD-33-120 utilizam-se os conversores A/D (analógico para digital) e D/A (digital para analógico) que a compõe servindo como interface entre a UM-33-100 e a placa de aquisição de dados MIC926 instalada no PC. Outras informações complementares sobre o Kit Feedback® – Servo Fundamentals Trainer estão disponíveis no site (www.fbk.com.br) e nos manuais da Feedback® (Treinamento dos Fundamentos Servo Analógico 33-002 e outros, e ainda Treinamento dos Fundamentos Servo Digital - 33-003 (STF154) e outros).

Na figura 1 são apresentados os equipamentos a serem utilizados no desenvolvimento já comunicando um com o outro. O protocolo de comunicação utilizado é um protocolo proprietário especificado pelo manual da placa de aquisição que compõe o Kit Feedback® – Servo Fundamentals Trainer.

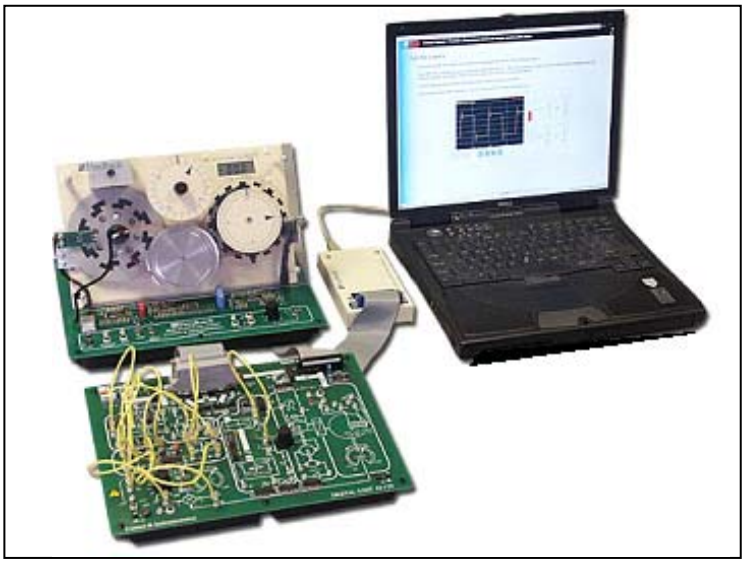

Figura 1 – UM-33-100 e a UD-33-120 e o PC instalados (FEEDBACK®)

A seguir, na seção 3 alguns trabalhos sobre modelagem de sistemas dinâmicos serão brevemente comentados relacionando-os com a proposta deste artigo.

3. Breve Estado da Arte

Essa seção destinada à revisão bibliográfica é apenas um estudo que se restringiu, a saber, a evolução das pesquisas sobre RNA's nos últimos anos principalmente nas aplicações de modelagem de sistema dinâmicos usando esse método. A preocupação em fazer uma pesquisa mais minuciosa sobre seus trabalhos não foi o foco.

Em sua dissertação de mestrado Vargas (1997) fez algumas investigações a respeito de algoritmos de treinamentos que obtivesse melhor desempenho em técnicas de identificação de sistemas dinâmicos não lineares através de RNA. Com isso ele comparou outras técnicas com a comumente usada (*backpropagation*). E a que mais foi focada em seus estudos foi as Redes Neurais Recorrentes com a implementação de um algoritmo de aprendizagem sob o contexto da teoria de estabilidade de Lyapunov. Como exemplo prático para verificar a eficiência de seus estudos ele fez a modelagem da difusão do serviço de TV a cabo nos Estados Unidos.

O intuito da verificação do trabalho de Vargas (1997) era de identificar outros possíveis métodos a serem testados para a modelagem do problema desse trabalho, sem sair do foco de RNA. Mas observou-se que a principio bastava implementar técnicas mais comuns e mais difundidas do que partir logo para uma solução através de técnicas mais avançadas, como foi o caso proposto por ele.

Após observarem que o difundido algoritmo de treinamento *backpropagation* não correspondia tão bem com sua eficiência na modelagem de sistemas dinâmicos não lineares quando o mesmo possui uma dinâmica linear interna, Carvalho e Oliveira (2004), implementaram o método de identificação de sistemas não lineares por RNA, usando Algoritmos Genéticos para a aprendizagem da mesma. Para comprovar que nesse caso o *backpropagation* não era o melhor recomendado eles também o utilizaram para fins de comparação.

Observou-se através do trabalho de Carvalho e Oliveira (2004) que em todos os casos, nem sempre o algoritmo *backpropagation* é o que proporciona um melhor resultado, outros algoritmos de aprendizagem (como por exemplo, Algoritmos Genéticos) dependendo da aplicação é a melhor opção. Uma verificação que Carvalho e Oliveira (2004) encontraram foi que em sistemas dinâmicos não lineares, mas que possuem dinâmica linear interna o *backpropagation* não é eficiente. Para este trabalho optou-se pelo *backpropagation* porque o sistema (motor CC) possui uma dinâmica não linear, porém tendendo a linearização e ainda pela familiarização com o mesmo.

Oliveira (2004) realizou a estimação dos parâmetros de um motor de indução trifásico através de Redes Neurais. O algoritmo de treinamento utilizado para esse sistema foi o Filtro de Kalman. A conclusão obtida foi que o sistema desempenhou resultados muito satisfatórios. O que comprova a eficiência desse método e com um custo computacional e de implementação bem baixo.

Vários estudos são feitos acerca de estimação de parâmetros, um motor de indução assim como os motores CC são importantes atuadores de máquinas e sistemas, por isso a necessidade de cada vez mais estudos aprofundados sobre os seus modelos. Pois o quesito otimização está intimamente ligado em eficiência, qualidade e economia de recursos e equipamento.

Furtado (2008), propôs uma comparação entre algumas técnicas de identificação de dados em sistema dinâmicos. Onde conclui que a RNA do tipo *Perceptron Multicamadas* teve um melhor desempenho que o Filtro de Kalman, Filtro de Partículas, e o Método Variacional. Todas essas técnicas são usadas em sistemas operacionais de previsão baseados em modelos físico-matemáticos. O sistema utilizado para a aplicação destas técnicas foi o sistema de Lorenz em regime caótico, que se caracteriza por ser um sistema dinâmico bastante crítico.

Diversas pesquisas concluíram que a RNA tem um ótimo desempenho ao se deparar com problema de características não linear. Furtado (2008) usou um exemplo de sistema caótico, porém não associou esse problema a uma aplicação prática. A proposta nesse artigo é usar um sistema também não linear, porém, um modelo real e ainda sugerindo suas aplicações.

Na seção seguinte os materiais e os métodos utilizados para o desenvolvimento desse trabalho serão comentados. A síntese do modelo de RNA será apresentada, assim como se chegou a ela.

4. Materiais e Métodos

Definiu-se anteriormente que o problema em questão é o de identificar um modelo matemático no padrão de RNA que se comporte de maneira mais próxima do sistema real, ou seja, estimação de parâmetros. E foi observado também que se tratava de um sistema dinâmico e que de acordo com a variação do tempo o sistema demonstra características inerentes a cada instante. Então, foi preciso fazer um levantamento prévio do comportamento dinâmico do sistema em questão. Para esse tipo de levantamento é comum que o sistema seja submetido a uma excitação em degrau. Que nada mais é que uma variação brusca na entrada, para verificar o seu comportamento na saída. A figura 2 ilustra a forma de coleta dos dados.

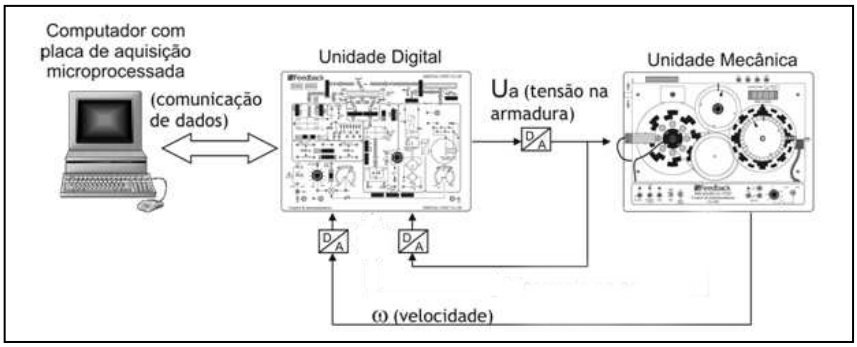

Figura 2 – Ilustração do esquema de coleta de dados (LEITE, KUGA & LOPES, 2006)

A princípio foram feitas vinte simulações com duração de 10 segundos cada. Em cada simulação o degrau era dado no instante de 5 segundos, variando o valor de tensão e observando na sua saída o comportamento da velocidade. O sensor de velocidade disponível é um tacogerador CC. O software utilizado para programação do experimento foi o MATLAB® com o Toolbox Real-Time Workshop, destinado a experimentos em tempo real. Os conversores A/D e D/A têm a mesma resolução (*i.e.* 8 bits) com o período de amostragem dos dados de 0,01s. Conforme já apresentado o motor CC utilizado trabalha numa faixa de -10V até 10V, então para a coleta de amostras os degraus variaram nessa mesma faixa unidade por unidade de tensão, sendo feito com números inteiros apenas por questão de redução de quantidade de amostras e tendência a linearidade do sistema.

Esses valores coletados foram armazenados no MATLAB® na forma de uma matriz A_{mxn} (onde m=1000 e n=2). Como essa matriz formava uma base de dados extensa sem variação significativa em cada instante e para a RNA não eram necessários valores tão próximos. Para o pré-tratamento dos dados amostrais antes de serem utilizados, foi desenvolvido um algoritmo que segundo Carvalho (2004) tem as seguintes funções: redução da matriz de dados; atrasos nas variáveis gerando novas entradas; normalização de acordo com as funções de ativação e divisão do conjunto inicial de amostras em conjunto treinamento e teste e validação.

Após serem tratados os dados a etapa seguinte foi a do projeto da RNA e uma dúvida era escolher qual tipo de rede deveria ser implementada. Como segundo Haykin (2001), para modelagem e identificação de sistemas dinâmicos o formato de redes neurais mais usadas são as de múltiplas camadas do tipo *feedforward* (alimentadas à diante) utilizando o algoritmo *backpropagation*. Pode-se dizer que o algoritmo *backpropagation* é uma generalização do método dos mínimos quadrados, que utiliza técnicas de gradiente descendente interativo para minimizar uma função de custo igual ao erro médio quadrado entre a saída desejada e a saída da RNA. Essa rede elaborada contém duas camadas: a camada oculta composta de três neurônios e usando a função de ativação do tipo *logsig* e a camada de saída com um neurônio e a função do tipo *pureline*. A figura 3 ilustra um exemplo da configuração da rede usada.

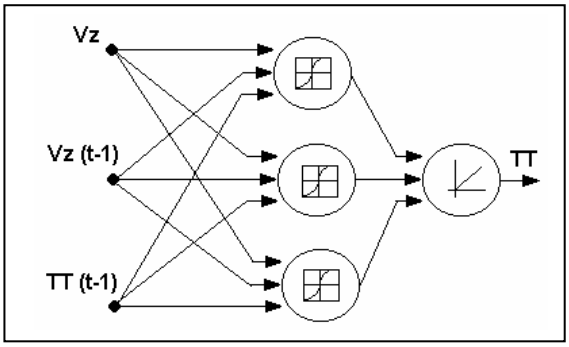

Figura 3 – Ilustração de uma RNA do tipo múltiplas camadas

Para cada tipo de RNA existe seu modelo matemático correspondente. O que representa e rede implementada nesse trabalho, com as características citadas anteriormente é o que segue,

$$N1 = \frac{1}{e^{-(\sum xi \cdot wij + bj)}}$$

$$N2 = \frac{1}{e^{-(\sum xi \cdot wij + bj)}} \qquad (2)$$

$$N3 = \frac{1}{e^{-(\sum xi \cdot wij + bj)}}$$

referente a camada oculta, e

$$N4 = y = \sum Nj \cdot wij + bj \qquad (3)$$

referente a camada de saída.

Onde *xi* é a *i-ésima* entrada, que comparada a figura 3 corresponde a Vz, Vz(t-1) e TT (t-1). *y* a saída, que corresponde a TT. *wij* é o peso de conexão entre a entrada *i-ésima* e o neurônio *j-ésimo*, ou seja, o peso de cada entrada em cada neurônio. *bj* é a bia no neurônio *j-ésimo*, ou seja, a bia em cada neurônio e N*j* é a saída de cada uma das funções de ativação no neurônio *j-ésimo*, ou seja, corresponde a saída de cada da função de ativação de cada neurônio. Sendo *i=1, 2, 3, 4* e *j=1, 2, 3, 4.*

Para o projeto da RNA optou-se pela utilização do software de simulação MATLAB®, onde nele encontra-se um *toolbox* apropriado para redes neurais (*nntool*). A escolha dele se justifica por ser uma poderosa e eficiente ferramenta, além de ter interface bastante "amigável" dando um grau de simplicidade extremamente importante ao usuário. Ele precisa apenas acessar essa interface no programa e configurar a RNA da forma mais conveniente.

Os parâmetros da RNA foram escolhidos da seguinte maneira: função treinamento Levenberg-Marquardt *(trainlm)*, pois apresenta bom desempenho em casos práticos fazendo a rede convergir mais rápida a um ponto ótimo. O índice de desempenho escolhido para acompanhar a eficiência da rede foi o MSE, por ter características aproximadas do algoritmo de treinamento escolhido. Sendo que o valor a ser usado como critério de parada é o mais próximo de zero desde que obedecido o número de épocas para a convergência *(epochs = 100)*. Os demais parâmetros foram *default*.

Após toda a configuração da RNA, o próximo passo é o treinamento da rede visando encontrar os valores de pesos e bias adequados para interagir com o sistema da maneira mais apropriada a trazer a minimização do erro. Foi utilizado o conjunto de dados pré-tratados para o treinamento. Definiram-se as entradas e o padrão de saída desejado. No instante inicial o próprio software arbitra valores para os pesos e bias para que eles sejam modificados quando treinados.

Os valores de pesos e bias encontrados são os apresentados na tabela a seguir:

	X1	*X2*	*X3*	*Bias*
N1	w_{11} = 5.7201	w_{21} = 5.5852	w_{31} = 7.7122	b_1 = -9.8313
N2	w_{12} = 6.2787	w_{22} = -3.5577	w_{32} = 3.1776	b_2 = -3.1282
N3	w_{13} = -11.9332	w_{23} = 27.0293	w_{33} = -7.8961	b_3 = -8.253
N4	w_{14} = 0.63992	w_{24} = 0.29615	w_{34} = 0.31384	b_4 = -0.012543

Tabela 1 – Valores encontrados de Pesos e bias para a RNA

Encontrados esses valores a etapa seguinte é validar o projeto usando os dados de teste e compilando em forma tabular e gráfica os MSE e as características de desempenho.

5. Resultados

Para a verificação dos resultados os dados de teste foram utilizados e confrontados com os valores estimados pela RNA através de gráficos. Cada experimento apresentado a seguir corresponde a uma base de dados ao se variar a amplitude do degrau. Como o sistema tem uma tendência à linearidade sua variação não era tanta entre cada um dos testes, optou-se por utilizar apenas os valores de degrau par *(i.e.* -10, -8, -6, -4 etc) , variando de -10V até +10V, porque esses são os limites máximos de tensão para o funcionamento do motor CC. Porém, antes de fazer a verificação dos conjuntos de teste foi confrontado o MSE do conjunto de treinamento e também do conjunto de teste para validação do projeto. Essa verificação foi feita com a amplitude de degrau de -1V. Observou-se que teve diferença, mas os dois valores estavam próximos de zero, podendo assim ser considerado satisfatório. Após a validação os demais experimentos com o conjunto de teste e validação foram feitos, resultando na tabela do MSE e nos gráficos da figura 4.

Degrau(V)	***Conjunto de dados***	***MSE***
-1V	Treinamento	1.0063e-004
-1V	Teste1	7.3994e-004
-10V	Teste2	6.3447e-004
-8V	Teste3	6.3549e-004
-6V	Teste4	4.1558e-004
-4V	Teste5	3.9008e-004
-2V	Teste6	5.1851e-004
2V	Teste7	0.0011
4V	Teste8	0.0016
6V	Teste9	0.0020
8V	Teste10	0.0038
10V	Teste11	0.0038

Tabela 2 – Valores do MSE para as diversas amplitudes de degrau

Os gráficos a seguir representam o desempenho do modelo estimado quando comparado à resposta do sistema real, para os diferentes degraus aplicados.

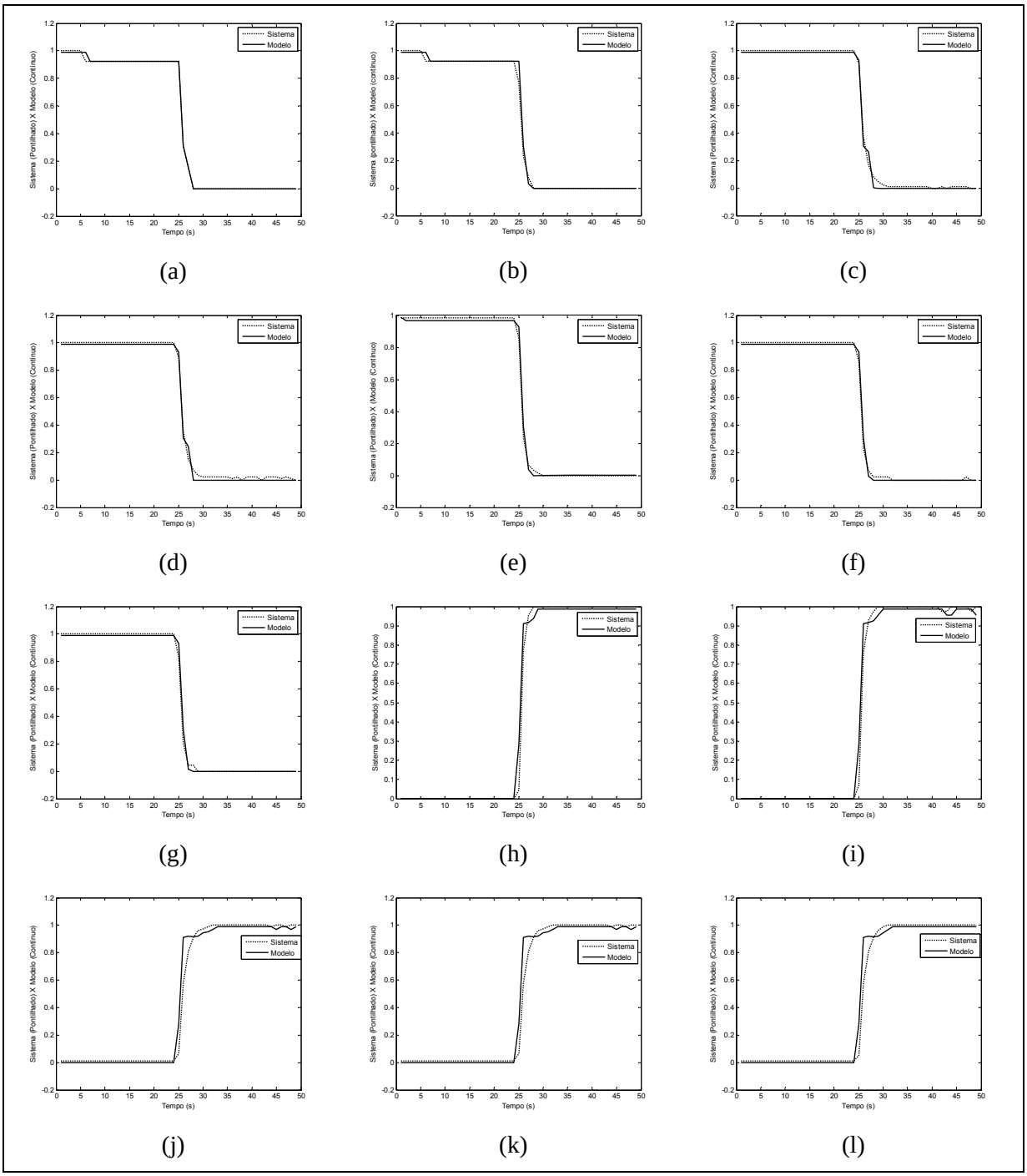

Figura 4 – Desempenho Modelo *versus* Sistema para os diferentes degraus de tensão. Onde (a) representa o degrau de -1V para o conjunto de treinamento e de (b) até (l) são para os conjuntos de teste variando os seguintes degraus de tensão respectivamente: -1V, -10 V, -8 V, -6 V, -4 V, -2 V, 2 V, 4 V, 6 V, 8 V e 10 V

Na próxima seção uma síntese das considerações e perspectivas para novas pesquisas sobre otimização e modelagem de sistemas dinâmicos, em especial o motores CC com o uso das RNA's será feita.

5. Considerações e Perspectivas

A modelagem do sistema em questão através da RNA apresentou um resultado considerado válido, com os MSE pequenos para diferentes valores de tensão aplicados como degrau. O *backpropagation*, para esse problema em questão obteve excelente resultado o que comprova em mais um exemplo sua eficiência como algoritmo de treinamento de RNA para modelagem de sistemas dinâmicos.

Uma vantagem de usar as RNA's para modelagem se comparada a métodos estatísticos de estimação (como os mínimos quadrados, por exemplo) se evidencia quando o sistema a ser estimado tem a característica de não ser linear ou existem muitas variáveis compondo o mesmo. Com essas características, fica muito mais trabalhoso e caro computacionalmente utilizar esses métodos. As RNA's obtêm vantagem por ser mais simples e eficiente. A sua forma de processamento é uma só, e não especificamente para cada sistema como é o caso dos métodos estatísticos. Uma sugestão para possivelmente trazer um resultado ainda melhor seria fazer testes mudando a quantidade de neurônios na camada oculta. Pois como já comentado o desempenho de uma RNA está ligado ao número de elementos nessa camada e não o número de camadas ocultas. Ao observar através da tabela dos MSE e também dos gráficos percebe-se que o modelo de RNA proposto apresentou um ótimo desempenho.

Para futuros trabalhos sugere-se utilizar as RNA's não só para estimação e sim para o otimização dos processos. Um exemplo seria o próprio motor CC, como sua utilidade para o meio industrial é grande, um sistema que trabalhe de maneira ótima se torna necessário.

Referências

Basilio, J.C. e Moreira, V.M. Experimentos para estimação dos parâmetros de motores de corrente contínua. *COBENGE – Congresso Brasileiro de Ensino de Engenharia*, 2001.

Carvalho, A.S. Identificação por RNA da relação vazão de entrada por temperatura de topo de uma coluna de destilação didática. *Monografia – Curso Superior de Tecnologia em Automação Industrial*, Centro Federal de Educação Tecnológica de Campos – CEFET Campos, Campos do Goytacazes, RJ. 2004, 82p.

Carvalho, F.G.M. de e Oliveira, R.C.L. de. Identificação de sistemas dinâmicos não-lineares utilizando Redes Neurais Artificiais treinadas com Algoritmos Genéticos. *CBA – Congresso Brasileiro de Automática*, 2004.

Caudill, M. *Expert network*. Byte. October. 1991. pp 108-116.

Corrêa, G. R. Desenvolvimento e implementação em hardware de um controlador neural. *Dissertação (Mestrado) – Departamento de Engenharia Elétrica*, Centro de Ciências Exatas, Universidade Federal de Uberlândia, Uberlândia, 1995.

Dorf, R. C. & Bishop, R. H. *Modern control systems*. 10th Ed. Hardcover, 2004.

Feedback, Instruments Ltda. *Analogue Servo - Fundamentals Trainer – 33-002*. Ed.03.

Feedback, Instruments Ltda. *Digital Servo - Fundamentals Trainer – 33-003 (SFT154)*. Ed.02.

Freeman, J. A. & Skapura, D. M. *Neural Network - algorithms and programming techniques*. Reading. Addison Wesley Pub. Comp. 1991.

Furtado, H. C. M. Redes neurais e diferentes métodos de assimilação de dados em dinâmica não linear. São José dos Campos: INPE, 2008.123p.(INPE-15235-TDI/1322)<http://mtc-m17.sid.inpe.br/col/sid.inpe.br/mtc-m17@80/2008/02.07.10.49/doc/publicacao.pdf> 11, 2009.

Haykin, S. *Redes Neurais – Princípios e práticas.* Porto Alegre: Bookman, 2001.

Leite, A. C. Kuga, H. K. & Lopes, R. V. F. Model-based fault detection on a reaction wheel. *Brazilian Conference on Dynamics*, Control and their Applications. Guaratinguetá, SP, Brasil, 2006.

Leonhard, W. *Control of electrical drives.* 3nd Edition, Springer-Verlag: Berlin, 2001.

Medeiros, J.S. Bancos de dados geográficos e Redes Neurais Artificiais: Tecnologias de apoio à gestão do território. *Dissertação (Doutorado) – Departamento de Geografia*, Universidade de São Paulo, São Paulo, 1999. http://www.dpi.inpe.br/teses/simeao/. 04, 2010.

Navarro, C. M. Projeto de um Controlador Adaptativo por Modelo de Referência (MRAC) para um disco rotativo diretamente acoplado ao eixo de um motor de corrente contínua (Unidades Feedback: 33-100 e 33-120). *Monografia – Curso Superior de Tecnologia em Automação Industrial,* Centro Federal de Educação Tecnológica de Campos – CEFET Campos, Campos do Goytacazes, RJ. 2008, 72p.

Oliveira, R. H. Estimação de parâmetros do motor de indução trifásico com o uso de redes neurais recorrentes. *Dissertação de mestrado* - Universidade Federal de Itajubá, MG. 2004.

Vargas, J.A.R. Identificação de sistemas dinâmicos via redes neurais artificiais. *Dissertação de mestrado* - Engenharia Eletrônica e Computação, Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, 1997. http://www.bd.bibl.ita.br/tde_busca/arquivo.php?codArquivo=273. 04, 2010.

WEG. *Motores de corrente contínua.* Catálogo, 2004.