

**COMPARAÇÃO ENTRE O MÉTODO DE MÍNIMOS QUADRADOS E REDES  
NEURAS ARTIFICIAIS PARA A ESTIMAÇÃO DE PARÂMETROS DE UM MOTOR  
DE CORRENTE CONTÍNUA**

**CLEBER DE MEDEIROS NAVARRO**

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE - UENF  
CAMPOS DOS GOYTACAZES - RJ  
JULHO - 2012

**COMPARAÇÃO ENTRE O MÉTODO DE MÍNIMOS QUADRADOS E REDES  
NEURAS ARTIFICIAIS PARA A ESTIMAÇÃO DE PARÂMETROS DE UM MOTOR  
DE CORRENTE CONTÍNUA**

**CLEBER DE MEDEIROS NAVARRO**

“Dissertação apresentada ao Centro de  
Ciência e Tecnologia, da Universidade  
Estadual do Norte Fluminense, como  
parte das exigências para a obtenção  
do título de mestre em Engenharia de  
Produção”.

ORIENTADOR: Prof. Geraldo Galdino de Paula Júnior, D. Sc.

CAMPOS DOS GOYTACAZES - RJ

JULHO - 2012

**COMPARAÇÃO ENTRE O MÉTODO DE MÍNIMOS QUADRADOS E REDES  
NEURAS ARTIFICIAIS PARA A ESTIMAÇÃO DE PARÂMETROS DE UM MOTOR  
DE CORRENTE CONTÍNUA**

**CLEBER DE MEDEIROS NAVARRO**

“Dissertação apresentada ao Centro de  
Ciência e Tecnologia, da Universidade  
Estadual do Norte Fluminense, como  
parte das exigências para a obtenção  
do título de mestre em Engenharia de  
Produção”.

Aprovada em \_\_\_\_\_ de \_\_\_\_\_ de 2012.

Comissão Examinadora:

---

Prof<sup>a</sup>. Jacqueline Magalhães Rangel Cortes - UENF

---

Prof. Adelson Siqueira Carvalho - IFF/campus Campos Centro

---

Prof. José Ramón Arica Chávez - UENF

---

Prof. Geraldo Galdino de Paula Junior, D. Sc. - UENF (Orientador)

**FICHA CATALOGRÁFICA**  
**Preparada pela Biblioteca do CCT/UENF 60/2012**

Navarro, Cleber de Medeiros

Comparação entre o método de mínimos quadrados e redes neurais artificiais para a estimação de parâmetros de um motor de corrente contínua / Cleber de Medeiros Navarro - Campos dos Goytacazes, 2012.

xvi, 88 f. : il.

Dissertação (Mestrado em Engenharia de Produção) – Universidade Estadual do Norte Fluminense Darcy Ribeiro. Centro de Ciência e Tecnologia. Laboratório de Engenharia de Produção.  
Campos dos Goytacazes, 2012.

Orientador: Geraldo Galdino de Paula Junior.

Área de Concentração: Pesquisa Operacional.

Bibliografia: f. 75-79.

1. MOTOR CC 2. SERVISSISTEMA 3. MODELAGEM DE SISTEMAS DINÂMICOS  
4. RNA 5. MÉTODO DE MÍNIMOS QUADRADOS I. Universidade Estadual do Norte Fluminense Darcy Ribeiro. Centro de Ciência e Tecnologia. Laboratório de Engenharia de Produção II. Título.

CDD 519.7

*Dedico este trabalho a todos que acreditam  
que o limite para qualquer coisa na vida é  
a morte.*

*A adversidade desperta em nós capacidades que, em circunstâncias favoráveis, teriam ficado adormecidas.*

*Horácio*

## **Agradecimentos**

À minha família, por estarem sempre ao meu lado, torcendo e me incentivando a nunca parar com os estudos e me dando forças para seguir de cabeça erguida, em especial meus pais pela paciência diante de estresses causados por noites mal dormidas; à minha mulher Viviane pelo apoio moral nesta etapa final do curso; ao meu ex, porém eterno professor, parceiro de trabalho e também amigo Adelson por todos os conselhos, incentivo e pela coorientação nesta dissertação. Sou eternamente grato por isso! Ao IF Fluminense pela estrutura, principalmente ao setor de automação industrial pelo suporte estrutural na realização dos estudos práticos para a elaboração desta dissertação; aos professores do Programa de Pós-Graduação em Engenharia de Produção da UENF pelo conhecimento adquirido e pelas informações que servem tanto para minha vida profissional quanto pessoal; meus sinceros agradecimentos em especial ao meu orientador e professor Geraldo Galdino por ter depositado confiança em mim na orientação, pela paciência, conselhos etc. Tenho certeza que sem a sua prestatividade esta dissertação teria um conteúdo bem menos aprimorado.

## RESUMO

O motor cc é o atuador mais popular entre os dispositivos mecatrônicos modernos, devido à sua simplicidade, alto torque na partida e em baixas rotações, facilidade de controlar a sua velocidade e vida útil bastante extensa sem manutenção. Para que servossistemas que utilizam o motor cc como atuador trabalhem de maneira eficiente, várias estratégias de controle são estudadas e desenvolvidas em diferentes linhas de pesquisa. Para maior eficácia desses sistemas de controle é preciso um estudo minucioso das variáveis envolvidas e dos equipamentos utilizados para modelar seus comportamentos. Esses estudos resultam em modelos matemáticos que refletem da melhor maneira possível o comportamento do equipamento. Com base nesses conceitos o presente trabalho apresenta a comparação de dois métodos de estimação de parâmetros de um motor cc: o método de Mínimos Quadrados e Redes Neurais Artificiais (RNA). O equipamento utilizado para os testes foi um motor cc do Kit Feedback® - Servo Fundamentals Trainer. Este é um kit didático que possibilita colocar em prática várias técnicas de controle. Com ele é possível simular o motor cc como um sistema dinâmico real, sem perder significativamente suas características de processo. Inicialmente foi feito o levantamento dos dados do motor cc que relaciona a resposta da velocidade dele ao degrau de tensão aplicado em seus terminais para o seu funcionamento. Vários valores para este degrau foram selecionados e os seus resultados armazenados numa matriz de dados. O modelo de RNA utilizado foi a do tipo feedforward multicamadas com aprendizado supervisionado pelo algoritmo de treinamento backpropagation e o método estatístico foi os Mínimos Quadrados Linear. O índice de desempenho MSE (Mean Square Error) foi analisado com a intenção de verificar a menor diferença entre a resposta do sistema real e o modelo estimado. A validação dos dois métodos foi feita de forma quantitativa. Os gráficos apresentados deram a possibilidade de observar a discrepância entre o modelo estimado e o sistema. Após a análise do desempenho pelos dois métodos propostos, sendo a RNA que melhor representava o motor cc é preciso enfatizar que a solução apresentada não é a definitiva nem ótima, outros métodos podem ser usados em conjunto ou individualmente com a finalidade de tentar otimizar ainda mais os resultados.

Palavras-chave: Motor cc, Servossistema, Modelagem de Sistemas dinâmicos, RNA, Método de Mínimos Quadrados.

## ABSTRACT

The dc motor is the most popular actuator among modern mechatronic devices due to its simplicity, high torque at startup and at low rotations, ease of controlling its speed and very long service life without maintenance. For using efficiently the servo dc motor as actuator, various control strategies are studied and developed on different lines of research. For maximum effectiveness of these control systems, it is needed a thorough study of the variables involved, and the equipment used to be able to model their behavior. These studies result in mathematical models that reflect the best possible behavior of the equipment. Based on these concepts, this article presents a comparison of two methods of estimating parameters of an engine dc: the method of Least Squares and Neural Networks (ANN). The equipment used for the tests was an engine dc kit Feedback <sup>®</sup> - Servo Fundamentals Trainer. This kit is a teaching tool that allows to put into practice various control techniques. It is possible to simulate the dc motor as a real dynamic system, without significant loss of process characteristics. Initially, it was made the survey of the data dc engine that lists the response of his speed to a step voltage, applied to its terminals for its operation. Various values for this step were selected and their results stored in a data array. The ANN model used was the feedforward multilayer supervised learning, with the backpropagation training algorithm, and the statistical method was the Linear Least Squares. The performance index MSE (Mean Square Error) was analyzed with the intention of verifying the smallest difference between the response of the real system and the estimated model. The validation of two methods was made in a quantitative way. The graphs shown, gave the opportunity to see the discrepancy between the estimated model and the real system. After the analysis of performance by the two methods proposed, being the RNA that best represented the engine dc, is necessary to emphasize that the solution presented is not final or optimal. Other methods can be used together or individually in order to try further optimize more results.

Keywords: dc motor, servo, Modelling of dynamical systems, RNA, Method of Least Squares.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Problema. . . . .	2
1.2	Objetivo. . . . .	2
1.3	Questão de pesquisa. . . . .	3
1.4	Relevância. . . . .	4
1.5	Organização da dissertação. . . . .	5
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>6</b>
2.1	Redes Neurais. . . . .	6
2.2	Redes Neurais Artificiais (RNA). . . . .	7
2.2.1	Tipos de funções de ativação. . . . .	10
2.2.2	Redes Perceptron multicamadas (MLP). . . . .	13
2.2.3	Aprendizado. . . . .	15
2.2.4	Backpropagation. . . . .	16
2.3	O Método dos Mínimos Quadrados. . . . .	20
2.3.1	Qualidade de ajuste. . . . .	24
<b>3</b>	<b>ESTADO DA ARTE</b>	<b>25</b>
<b>4</b>	<b>KIT FEEDBACK® - SERVO FUNDAMENTALS TRAINER</b>	<b>31</b>

4.1	Unidade Mecânica-33-100. . . . .	31
4.2	Unidade Digital-33-120. . . . .	32
4.3	Esquema de comunicação. . . . .	33
4.4	Motor cc. . . . .	34
4.5	Sensor de velocidade (tacogerador). . . . .	38
4.6	Conversor Analógico-Digital (A/D). . . . .	39
4.7	Conversor Digital-Analógico (D/A). . . . .	39
<b>5</b>	<b>METODOLOGIA E IMPLEMENTAÇÃO</b>	<b>41</b>
5.1	Funcionamento do kit. . . . .	41
5.2	Coleta de dados. . . . .	42
5.3	Desenvolvimento da RNA. . . . .	46
5.3.1	Tratamento dos dados. . . . .	47
5.3.2	Criação da rede. . . . .	48
5.3.3	Configuração da rede. . . . .	49
5.3.4	Treinamento da rede. . . . .	51
5.3.5	Validação da rede. . . . .	54
5.4	Obtenção do modelo discreto do motor cc. . . . .	54
5.5	Estimação pelo Método dos Mínimos Quadrados. . . . .	55
<b>6</b>	<b>ANÁLISE DOS RESULTADOS</b>	<b>57</b>
6.1	Topologia da RNA escolhida. . . . .	57
6.2	Resultado da modelagem via RNA. . . . .	58
6.3	Resultado da estimação pelo Método dos Mínimos Quadrados. . . . .	64
<b>7</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>72</b>

7.1	Conclusão. . . . .	72
7.2	Perspectivas futuras. . . . .	73
	<b>Referências Bibliográficas</b>	<b>75</b>
<b>A</b>	<b>INSTALAÇÃO DO KIT E COMUNICAÇÃO COM O MATLAB®</b>	<b>80</b>
A.1	Ligação da UD-33-120. . . . .	80
A.2	Configuração do Simulink para comunicar com o kit. . . . .	81
A.3	Criação de diagrama de blocos. . . . .	84
A.4	Comunicação. . . . .	88

## Lista de Tabelas

4.1	Variáveis envolvidas em um motor cc . . . . .	37
5.1	Detalhes técnicos da coleta dos dados (adaptado de Leite (2007)) . . . .	44
5.2	MSE para os diferentes números de neurônios na camada oculta e diferentes estímulos de entrada. . . . .	52
5.3	Média geométrica do MSE para cada valor de neurônios na rede. . . . .	53
6.1	Valores encontrados de Pesos e bias para a RNA (Fonte: Navarro et al. (2010)). . . . .	58
6.2	Valores do MSE para as diversas amplitudes de degrau. . . . .	59
6.3	Parâmetros encontrados para cada degrau. . . . .	65
6.4	MSE encontrados pelos Mínimos Quadrados. . . . .	66
7.1	Comparação dos valores do MSE pelos dois métodos propostos. . . . .	73

## Lista de Figuras

2.1	Modelo de um neurônio natural. (Fonte: Ballone (2008)) . . . . .	7
2.2	Neurônio artificial proposto por McCulloch & Pitts (1943) . . . . .	8
2.3	Representação de um neurônio artificial genérico . . . . .	9
2.4	Função linear . . . . .	10
2.5	Função rampa . . . . .	11
2.6	Função degrau . . . . .	12
2.7	Função sigmoideal . . . . .	13
2.8	Modelo de uma rede MLP . . . . .	14
2.9	Aprendizado supervisionado . . . . .	15
2.10	Aprendizado não-supervisionado . . . . .	16
2.11	Fases do algoritmo backpropagation . . . . .	17
2.12	Dispersão dos dados e a reta de ajuste pelos Mínimos Quadrados . . . .	21
4.1	UM - 33-100. (Fonte: Feedback Instruments Ltda. (1995b)) . . . . .	32
4.2	UD - 33-120. (Fonte: Feedback Instruments Ltda. (1995b)) . . . . .	33
4.3	Kit Feedback instalado. (Fonte: Feedback Instruments Ltda. (1995b)) . .	34
4.4	Esquema de comunicação entre a UM-33-100, a UD-33-120 e o PC. . . .	34
4.5	Motor cc. (Fonte: Action Technology (2011)) . . . . .	35

4.6	Detalhamento das variáveis do motor cc em questão (Adaptado de Dorf & Bishop (2004)). . . . .	36
4.7	Diagrama de blocos da função de transferência simplificada do motor cc	37
4.8	Diagrama de blocos da função de transferência do tacogerador. . . . .	39
5.1	Visão geral do sistema em diagrama de blocos. . . . .	41
5.2	Ilustração do esquema de coleta de dados (Fonte: Leite et al. (2006)) .	43
5.3	Esquema de coleta de dados em diagrama de blocos. . . . .	44
5.4	Identificação do Offset no kit. . . . .	45
5.5	Passos para o desenvolvimento de uma RNA . . . . .	47
5.6	Janela do nntool . . . . .	49
5.7	janela de configuração RNA . . . . .	50
5.8	RNA do tipo MLP (adaptado de Carvalho et al. (2004)) . . . . .	53
6.1	Modelo RNA versus Sistema real (degrau -10V). . . . .	59
6.2	Modelo RNA versus Sistema real (degrau -8V). . . . .	60
6.3	Modelo RNA versus Sistema real (degrau -6V). . . . .	60
6.4	Modelo RNA versus Sistema real (degrau -4V). . . . .	61
6.5	Modelo RNA versus Sistema real (degrau -2V). . . . .	61
6.6	Modelo RNA versus Sistema real (degrau 2V). . . . .	62
6.7	Modelo RNA versus Sistema real (degrau 4V). . . . .	62
6.8	Modelo RNA versus Sistema real (degrau 6V). . . . .	63
6.9	Modelo RNA versus Sistema real (degrau 8V). . . . .	63
6.10	Modelo RNA versus Sistema real (degrau 10V). . . . .	64
6.11	Experimento pelo Simulink . . . . .	65
6.12	Modelo versus Sistema real (degrau -10V). . . . .	67

6.13 Modelo versus Sistema real (degrau -8V). . . . .	67
6.14 Modelo versus Sistema real (degrau -6V). . . . .	68
6.15 Modelo versus Sistema real (degrau -4V). . . . .	68
6.16 Modelo versus Sistema real (degrau -2V). . . . .	69
6.17 Modelo versus Sistema real (degrau 2V). . . . .	69
6.18 Modelo versus Sistema real (degrau 4V). . . . .	70
6.19 Modelo versus Sistema real (degrau 6V). . . . .	70
6.20 Modelo versus Sistema real (degrau 8V). . . . .	71
6.21 Modelo versus Sistema real (degrau 10V). . . . .	71
A.1 Conexões da UD-33-120. . . . .	81
A.2 Abrindo um arquivo novo. . . . .	82
A.3 Modificando para simulação externa. . . . .	82
A.4 Acessando opções de comunicação. . . . .	82
A.5 Selecionando drive do kit. . . . .	83
A.6 Configurando os parâmetros de simulação I. . . . .	83
A.7 Configurando os parâmetros de simulação II. . . . .	84
A.8 Abrindo bloco de comunicação do kit I. . . . .	84
A.9 Abrindo bloco de comunicação do kit II. . . . .	85
A.10 Abrindo bloco de comunicação do kit III. . . . .	85
A.11 Inserindo os blocos dos conversores A/D e D/A no arquivo. . . . .	86
A.12 Configurando os blocos dos conversores A/D e D/A. . . . .	86
A.13 Montagem de um diagrama de blocos básico. . . . .	87
A.14 Gerando código “C” e compilando o arquivo. . . . .	88
A.15 Dando inicio a comunicação. . . . .	88

# Capítulo 1

## Introdução

Os motores de corrente contínua (cc) têm grande utilidade no desenvolvimento de servossistemas (LEONHARD, 2001) devido a sua robustez, alto torque na partida e em baixas rotações, e facilidade de controlar a sua velocidade, entre outras vantagens. Têm grande aplicação no meio industrial, como em ferramentas de avanço, bobinadeiras, máquinas têxteis, guinchos e guindastes, exaustores (WEG, 2004), e ainda extrema utilidade em manipuladores robóticos e mecatrônicos (LEITE, 2007). Diversos métodos de estimação de parâmetros e controle desses motores são estudados e desenvolvidos, com a finalidade de otimização do sistema ao qual é aplicado. Um modelo mal estimado origina intrinsecamente uma baixa eficiência do sistema. Neste projeto procura-se comparar dois processos de estimação de parâmetros de um motor de corrente contínua: o método de mínimos quadrados e redes neurais artificiais.

Sabe-se que a estratégia de controle de processos tem a finalidade de otimizar um sistema com base em especificações pré determinadas no instante de um projeto (NAVARRO & LEITE, 2010). Ou seja, ela 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.

Na pesquisa operacional, no contexto de redes neurais artificiais e na estatística, modelos destinam-se ao auxílio na tomada de decisão. Suas características definem, na prática, o grau de representação e de melhoria que se deseja no processo de otimização dos sistemas. Por isso, deve-se ter atenção no processo de modelagem pois, qualquer estudo que depende dos dados estimados, pode não refletir a realidade eficientemente.

## 1.1 Problema.

A problema em questão destina-se a obter a modelagem do motor cc estimando seus parâmetros através de dois métodos: Mínimos Quadrados, usando a abordagem clássica da estatística para estimar esses parâmetros e as Redes Neurais Artificiais com embasamento na computação natural ou inteligência artificial.

Segundo Kuga (2005) estimar parâmetros significa estimar dados (valores) que não variam ao longo do processo de estimação. Para se fazer isso, necessita-se claramente de uma base de dados medidos que seja relacionada a esses parâmetros. Com base nesse raciocínio, é preciso modelar como esses parâmetros estão relacionados com os dados medidos. Ou seja, encontrar um modelo matemático do sistema que se aproxima ao máximo o comportamento real. Dorf & Bishop (2004) sintetizam de forma clara o processo de modelagem de sistemas dinâmicos, principalmente o motor cc. Outras fontes de informação para esse tipo de processo pode ser encontrada em Aguirre (2004), Coelho & Coelho (2004) e Leite (2007).

A inteligência artificial procura criar sistemas inteligentes que reproduzam aspectos do comportamento humano, tais como percepção, raciocínio, adaptação e aprendizado.

A inteligência artificial é o ramo do conhecimento que se propõe a entender e a construir entidades inteligentes artificiais que apresentem as mesmas capacidades das entidades inteligentes encontradas no mundo real, ou próximas a ela.

Segundo Juvinao Carbono (2001) a computação natural compreende uma grande variedade de sub-áreas empregadas na otimização de problemas, dentre elas: fractais, simulated annealing, lógica nebulosa, redes neurais artificiais e a computação evolucionária.

## 1.2 Objetivo.

O objetivo deste trabalho é fazer uma comparação entre a eficiência na estimação de parâmetros de modelagem de um motor cc pelo Método dos Mínimos Quadrados e por meio de Redes Neurais Artificiais.

O Método dos Mínimos Quadrados é bastante difundido na pesquisa científica. Sua origem, segundo Aguirre (2004), vem desde a época de Gauss quando o mesmo o formulou para processar observações astronômicas de corpos celestes. Segundo Coelho & Coelho (2004) uma das vantagens de usar esse método é que ele se adequa bem aos dados quando o processo em estudo apresenta um baixo nível de ruído. Já uma desvantagem é que ele não é adequado em sistemas não lineares. Formalmente ele trata de minimizar a soma do quadrado dos erros, ou seja, a diferença da resposta de um sistema a uma excitação pela resposta do modelo estimado deste sistema. Ajustando os dados da melhor maneira possível.

Segundo De Castro (2001), as redes neurais artificiais são inspiradas na estrutura do cérebro humano e têm o objetivo de apresentar características similares a ele, tais como: aprendizado, associação, generalização e abstração. Elas são compostas por diversos elementos, como os processadores (ou neurônios artificiais), altamente interconectados, que efetuam um número pequeno de operações simples e transmitem seus resultados aos processadores vizinhos.

Devido à sua estrutura, as redes neurais artificiais são bastante efetivas no aprendizado de padrões a partir de dados não-lineares, incompletos, com ruídos e até mesmo compostos de exemplos contraditórios. Algumas de suas aplicações são reconhecimento de padrões (imagem, texto e voz), previsão de séries temporais e modelagem de sistemas.

### 1.3 Questão de pesquisa.

A identificação de um processo pode ser explicada de uma forma simplificada, como o “conhecimento” da relação entre suas entradas e saídas. É amplamente utilizada na Engenharia, a fim de se realizar ensaios e experimentos mais facilmente e com menores custos, além de previsão e/ou compreensão do comportamento de processos.

A maioria dos sistemas reais (não lineares, multivariáveis, variáveis no tempo) apresenta uma complexidade elevada, o que acaba por tornar as técnicas mais convencionais de modelagem muito penosas. Por isso, segundo Dorf & Bishop (2004) para fazer aproximações e torná-los sistemas lineares é comum em métodos tradicionais

de estimação que se desprezem alguns parâmetros. Outra solução é a utilização de técnicas de identificação avançadas, a maioria delas baseada em inteligência computacional e o uso de metaheurísticas.

As metaheurísticas são métodos de busca destinados a encontrar uma boa solução, eventualmente a ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada, a qual tem que ser modelada para cada problema específico (SOUZA, 2003a).

Contrariamente às heurísticas convencionais, as metaheurísticas são de caráter geral e têm condições de escapar de ótimos locais. Com a reunião de conceitos das áreas de Otimização e Inteligência Artificial, viabiliza-se a construção das metaheurísticas.

Dentre os procedimentos enquadrados como metaheurísticas que surgiram ao longo das últimas décadas, destacam-se: RNA, Algoritmos Genéticos, Simulated Annealing, Busca Tabu, GRASP, VNS, Colônia de Formigas, etc.

A questão levantada é como as redes neurais e o método de Mínimos Quadrados podem ser úteis na identificação do modelo de um motor cc. Em seguida a comparação dos dois métodos será feita com a intenção de identificar qual método produz solução de forma viável em termos de simplicidade, exatidão e custo de implementação.

#### 1.4 Relevância.

Uma das motivações de se identificar modelos de sistemas dinâmicos se deve ao fato de que testes caros com o sistema real podem ser poupados. Pois diversos experimentos podem ser feitos apenas com o modelo estimado evitando expor o equipamento à possibilidade de danos em laboratórios ao fazer testes primários. A utilização de modelos é muito comum em simulação para validação de sistemas de controle, pois se ocorrer alguma falha no projeto não danificaria o equipamento. Porém, um modelo mal estimado afetaria na fidelidade da simulação. Outra vantagem de se fazer simulações com modelos é que robustez a falhas não poderiam ser testadas com o sistema real.

## 1.5 Organização da dissertação.

Esta dissertação está estruturada em 7 capítulos e os conteúdos referentes aos mesmos segue da seguinte maneira:

No capítulo 2 é apresentado o referencial teórico do trabalho. Inicialmente a atenção será reservada para abordar os conceitos de Redes Neurais Artificiais, mais precisamente as do tipo multicamadas. Esses conceitos focam-se mais a aplicação de modelagem e ajustes de funções. Em seguida, os conceitos sobre o método dos Mínimos Quadrados são apresentados, onde é focada a parte de modelos lineares.

No capítulo 3, é apresentada uma revisão bibliográfica abordando uma evolução cronológica dos trabalhos que aplicam as metodologias aqui sugeridas. Ou seja, trabalhos que focaram em aplicações tanto com RNA's quanto com o método dos Mínimos Quadrados aplicados a sistemas dinâmicos.

No capítulo 4, os materiais utilizados para o desenvolvimento da dissertação é apresentado. Primeiramente o kit didático usado para os estudos é comentado e, em seguida, o motor cc que é parte integrante desse kit será detalhado.

No capítulo 5 é apresentada a metodologia relativa ao desenvolvimento do trabalho proposto. Primeiramente é descrita a forma como foi feita a coleta de dados, para que os mesmos sejam aplicados pelos dois métodos. Em seguida, são comentados respectivamente a forma como se faz a modelagem por meio de RNA e pelo método dos Mínimos Quadrados.

No capítulo 6, após a metodologia ter sido descrita, os resultados serão apresentados. Todos esses resultados serão compilados numa tabela e ainda os gráficos de desempenho do modelo escolhido também serão apresentados.

No capítulo 7, são apresentadas as conclusões do trabalho além de sugestões para trabalhos futuros.

## Capítulo 2

### REFERENCIAL TEÓRICO

#### 2.1 Redes Neurais.

O cérebro humano é um processador natural que possui bilhões de neurônios. Estes que são unidades básicas do cérebro, isoladamente podem ser considerados análogos a uma unidade de processamento, pois aceita e combina estímulos de vários outros neurônios (muitas entradas), porém possui somente uma saída. Segundo Beale & Jackson (1992) os neurônios estão conectados uns aos outros através de sinapses, e juntos formam uma grande rede, chamada Rede Neural. As sinapses transmitem estímulos através de diferentes concentrações de  $\text{Na}^+$  (Sódio) e  $\text{K}^+$  (Potássio), e o resultado disto pode ser estendido por todo o corpo humano. Esta grande rede proporciona uma fabulosa capacidade de processamento e armazenamento de informação.

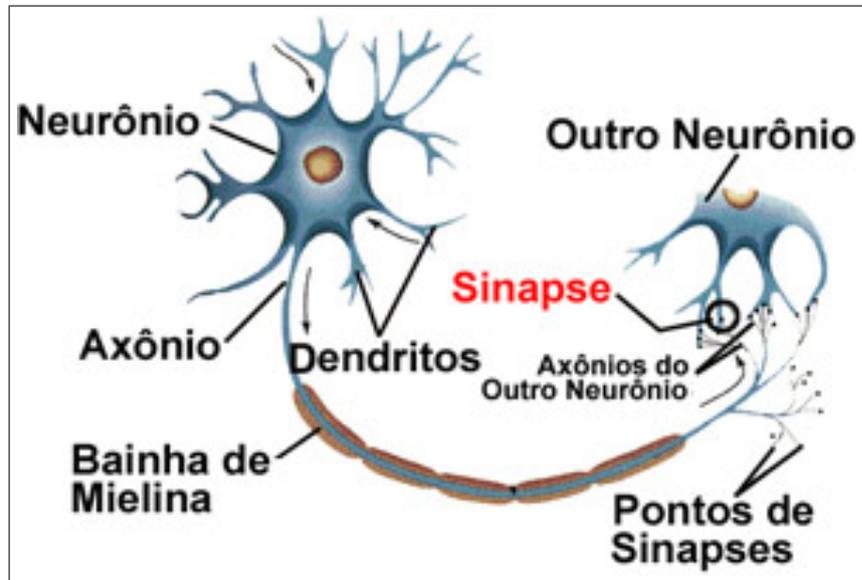
Segundo Freeman & Skapura (1991) os principais componentes dos neurônios são:

- Os dendritos, que tem por função, receber os estímulos transmitidos pelos outros neurônios;
- O corpo de neurônio, também chamado de soma, que é responsável por coletar e combinar informações vindas de outros neurônios;
- E finalmente o axônio, que é constituído de uma fibra tubular que pode alcançar até alguns metros, e é responsável por transmitir os estímulos para outras células.

Na figura 2.1 é ilustrado um modelo de um neurônio natural.

Ainda de acordo com Freeman & Skapura (1991) nos neurônios a comunicação é realizada através de impulsos, quando um impulso é recebido, o neurônio o processa, e passado um limite de ação, dispara um segundo impulso que produz uma substância neurotransmissora que flui do corpo celular para o axônio (que por sua

vez pode ou não estar conectado a um dendrito de outra célula). O neurônio que transmite o pulso pode controlar a frequência de pulsos aumentando ou diminuindo a polaridade na membrana pós sináptica. Eles têm um papel essencial na determinação do funcionamento, comportamento e do raciocínio do ser humano.



**Figura 2.1** Modelo de um neurônio natural. (Fonte: Ballone (2008))

Após este breve comentário sobre Redes Neurais, a próxima seção especifica melhor o assunto dando mais ênfase a RNA bem como ilustra o modelo de RNA proposto por McCulloch & Pitts (1943).

## 2.2 Redes Neurais Artificiais (RNA).

Uma RNA 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) e que segundo Braga et al. (2000) adquirem conhecimento através da experiência. Segundo Corrêa (1995) a principal finalidade das RNA's é explorar e reproduzir tarefas humanas naturais em processamento de informações. Além disso, elas também

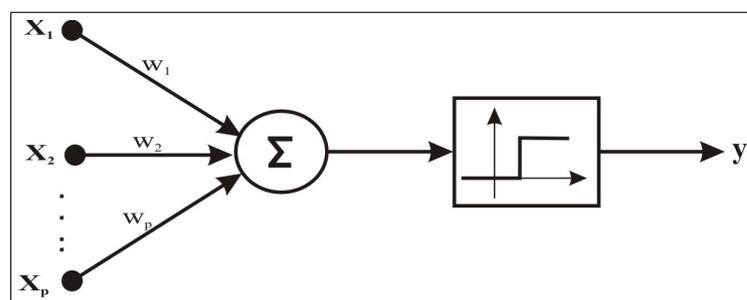
são usadas para solução de problemas de otimização combinatória, modelagem de sistemas, controle e aproximação de funções. De maneira geral, uma RNA tenta “simular” o sistema nervoso humano em forma de modelos matemáticos e códigos computacionais. Segundo Haykin (2001), ela se assemelha com o cérebro 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.

De forma geral, a operação de uma célula da rede se resume em:

1. Sinais apresentados na entrada;
2. Cada sinal é multiplicado por um peso que indica sua influência na saída da unidade;
3. É feita a soma ponderada dos sinais que produz um nível de atividade;
4. Se este nível exceder um limite “threshold” a unidade produz uma saída.

A figura 2.2, proposta por McCulloch & Pitts (1943) ilustra melhor o que foi comentado.

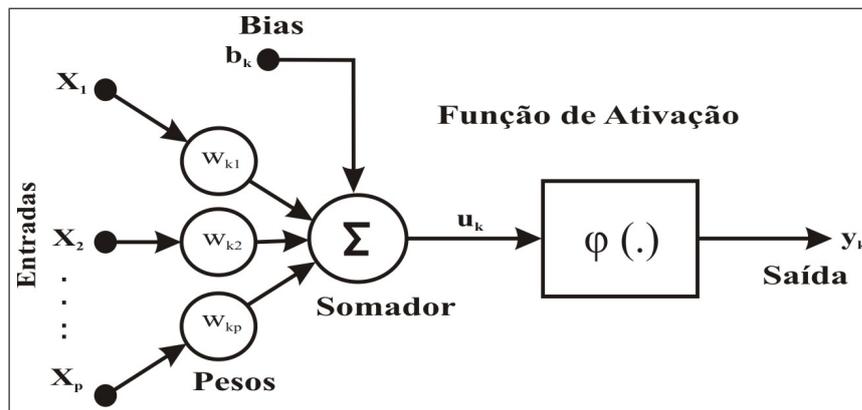


**Figura 2.2** Neurônio artificial proposto por McCulloch & Pitts (1943)

O neurônio proposto por McCulloch & Pitts (1943) foi aprimorado por Rosenblatt (1958) dando origem ao que é conhecido nos dias atuais como modelo dos Perceptrons. Inicialmente tinha como interesse o reconhecimento de padrões.

Atualmente, o modelo de neurônio artificial utilizado é denominado de neurônio genérico. A função de ativação não possui mais o limiar (threshold). Foi acrescentado um limiar (bias) que representa o peso associado a um sinal de entrada de valor unitário. Este limiar possui o efeito de aumentar (se for positivo) ou diminuir (se for negativo) a entrada líquida da função de ativação. Além disto, as entradas passaram a ser reais, e não mais apenas binárias, como no modelo de McCulloch & Pitts (1943).

A Figura 2.3 mostra a representação de um neurônio genérico.



**Figura 2.3** Representação de um neurônio artificial genérico

Matematicamente, a saída do neurônio genérico pode ser descrita como:

$$y_k = \varphi\left(\sum_{j=1}^p (w_{kj} \cdot x_j + b_k)\right) \quad (2.1)$$

onde  $x_1, \dots, x_p$  são sinais de entrada;  $w_{k1}, \dots, w_{kp}$  são os pesos sinápticos do neurônio  $k$ ;  $u_k$  é a saída do combinador linear devido aos sinais de entrada;  $b_k$  é o bias;  $\varphi(\cdot)$  é a função de ativação e  $y_k$  é o sinal de saída do neurônio.

O neurônio artificial representa uma aproximação extremamente simplificada do neurônio biológico. Para dar viabilidade a sua implementação computacional algumas das características básicas são conservadas.

Na subseção seguinte alguns tipos de funções de ativação mais usadas serão apresentadas.

### 2.2.1 Tipos de funções de ativação.

A função  $\varphi(\cdot)$  representa a função de ativação, a qual restringe a amplitude do sinal na saída de um neurônio. O papel da função de ativação é simular características não lineares de um neurônio biológico. A seguir, identificam-se quatro tipos básicos de função de ativação:

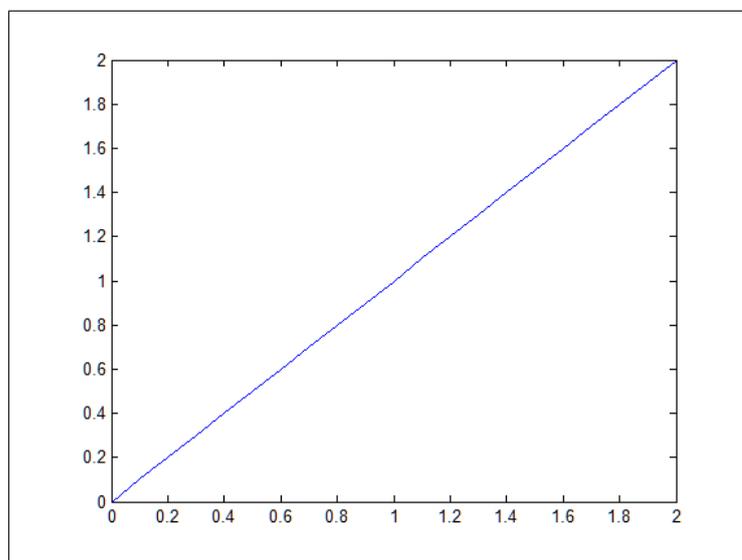
- Função linear

É definida pela equação:

$$y = \alpha x \quad (2.2)$$

onde  $\alpha$  é um número real genérico que define a saída linear  $y$  para os valores de entrada  $x$ .

A Figura 2.4 mostra o gráfico da função linear.



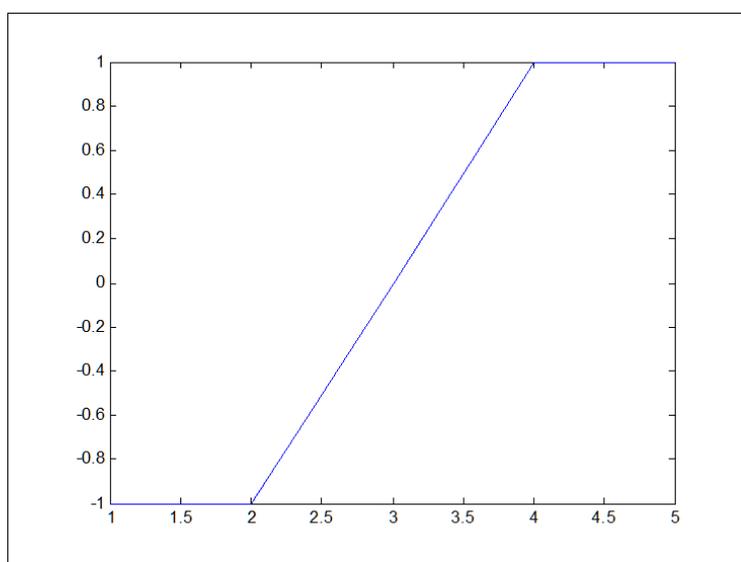
**Figura 2.4** Função linear

- Função rampa

A função rampa é uma restrição da função linear. Ela passa a ser uma função rampa quando esta restrição é feita para que somente produza valores constantes em uma faixa  $[-\gamma, +\gamma]$ . O seu valor mínimo é igual a  $-\gamma$  e máximo igual a  $+\gamma$ . Veja (2.3):

$$y = \begin{cases} +\gamma, & \text{se } x \geq +\gamma \\ x, & \text{se } \|x\| < +\gamma \\ -\gamma, & \text{se } \leq -\gamma \end{cases} \quad (2.3)$$

A Figura 2.5 mostra o gráfico da função rampa.



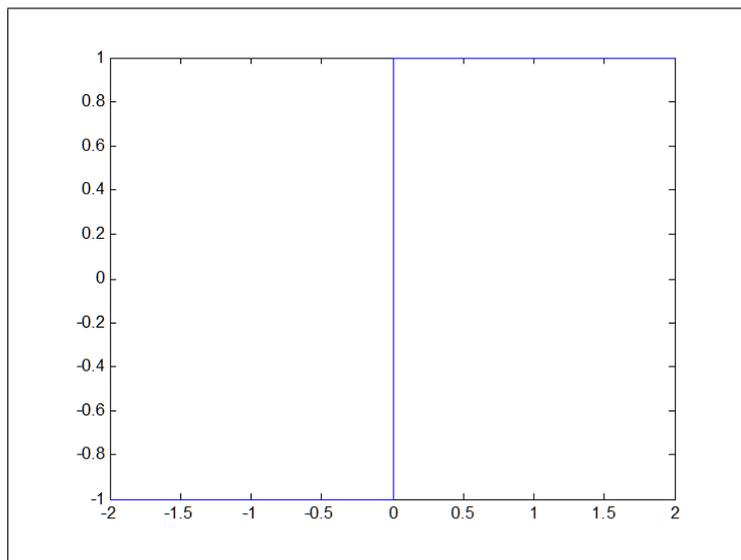
**Figura 2.5** Função rampa

- Função degrau

Essa função é similar a função sinal, pois produz uma saída  $+\gamma$  para valores de  $x$  maiores que zero, e  $-\gamma$  para valores menores do que zero. Veja (2.9):

$$y = \begin{cases} +\gamma, & \text{se } x > 0 \\ -\gamma, & \text{se } x \leq 0 \end{cases} \quad (2.4)$$

A Figura 2.6 mostra o gráfico da função degrau.



**Figura 2.6** Função degrau

- Função sigmoidal

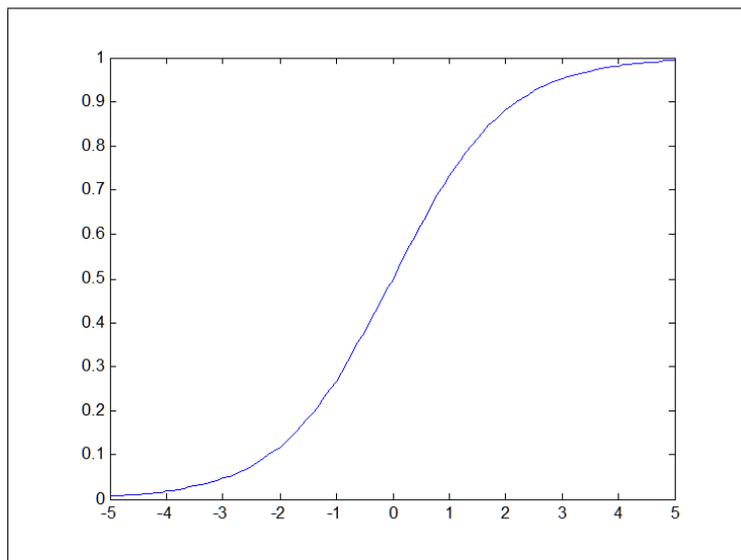
Também conhecida como *S-shape*, a função sigmoidal é uma função monotônica, semi-linear e limitada. É possível se definir várias sigmoidais distintas. São encontradas na construção de modelos das mais diversas áreas. Um exemplo deste tipo de função é a chamada função logística, definida na forma:

$$y = \frac{1}{1 + e^{-\alpha v}} \quad (2.5)$$

Onde  $\alpha$  é o parâmetro de inclinação da função sigmóide e  $v$  é o valor de ativação do neurônio.

A Figura 2.7 mostra o gráfico da função sigmoidal.

Existem vários modelos de RNA's usados atualmente. A utilizada nesta dissertação para resolver o problema de modelagem é a Rede Perceptron Multicamadas (MLP). Então, para melhor conhecimento sobre a mesma será comentado mais de rede MLP na subseção seguinte.



**Figura 2.7** Função sigmoideal

### 2.2.2 Redes Perceptron multicamadas (MLP).

Segundo Cerqueira (2001) a característica importante do modelo de rede perceptron foi à apresentação de um algoritmo de aprendizagem capaz de adaptar os pesos internos do neurônio de maneira que seja capaz de resolver o problema de classificação linear, em caso da separabilidade linear das classes.

O caso exemplar das limitações do perceptron é o problema “Ou exclusivo” (XOR) que prova que uma função tão simples de classificação não pode ser calculada pelo perceptron.

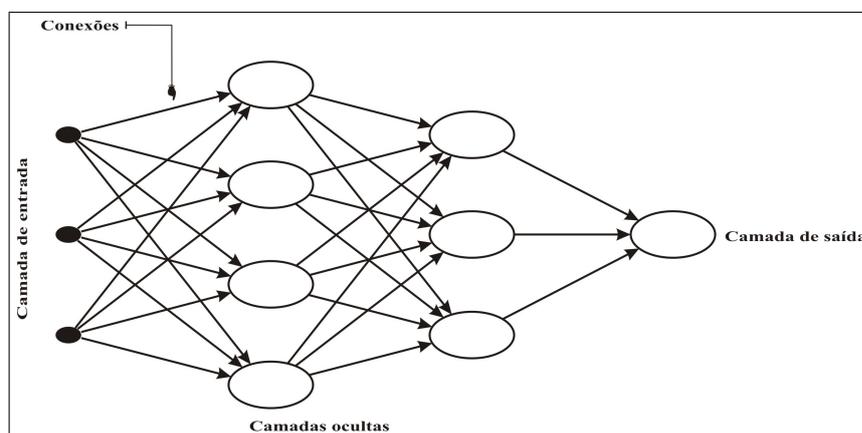
A solução para o problema XOR já era conhecida. Bastava acrescentar mais uma camada de neurônios na rede (uma camada escondida). O que faltava era um algoritmo que fosse capaz de treinar os pesos dessa rede multi-camada para que pudesse classificar corretamente problemas mais complexos.

O Multilayer Perceptron (MLP) foi concebido para resolver problemas mais complexos, os quais não poderiam ser resolvidos pelo modelo de neurônio básico. Os neurônios internos são de suma importância na rede neural, pois provou-se que sem estes torna-se impossível à resolução de problemas linearmente não separáveis.

Usualmente as camadas são classificadas em três grupos:

1. Camada de Entrada: onde os padrões são apresentados à rede;
2. Camadas Intermediárias ou Ocultas: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
3. Camada de Saída: onde o resultado final é concluído e apresentado.

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 & 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. A figura 2.8 ilustra uma rede MLP.



**Figura 2.8** Modelo de uma rede MLP

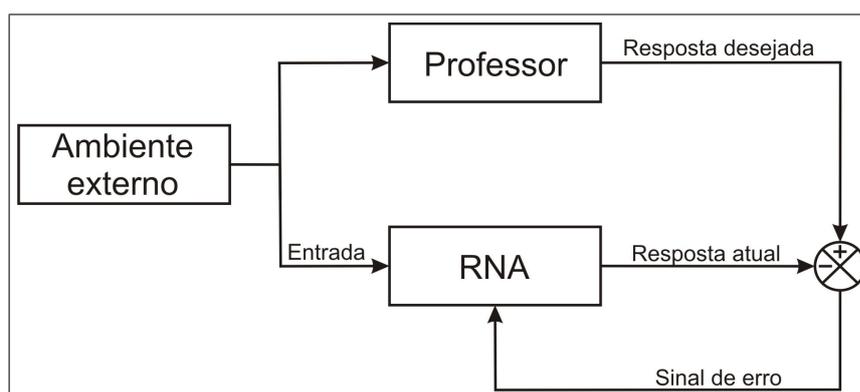
RNA's são também classificadas de acordo com a arquitetura em que foram implementadas, topologias, características de seus nós e regras de aprendizado. Com base nisso, a próxima subseção apresenta maiores esclarecimentos a respeito da classificação de uma rede conforme o aprendizado dela.

### 2.2.3 Aprendizado.

Como comentado na seção anterior, existem diversos tipos de RNA's e diferentes maneiras de classificá-las. Talvez a mais importante seja quanto à forma de aprendizado.

O processo de aprendizagem é realizado por meio de treinamento. Podendo esse ser caracterizado como aprendizado supervisionado ou não-supervisionado. 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.

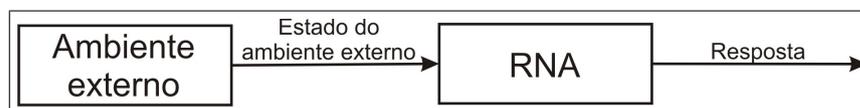
Segundo Freeman & Skapura (1991) o que define o tipo de aprendizado supervisionado é que nesse modelo são apresentadas à rede padrões de entrada e seus correspondentes padrões de saída. Assim, a rede ajusta os pesos das conexões seguindo uma determinada lei de aprendizagem predeterminada. Isso é feito até que o desvio entre os padrões de saída gerados pela rede e os padrões de saída medidos do sistema atinja um valor mínimo tolerado. Ainda segundo Freeman & Skapura (1991), existem dezenas de leis de aprendizagem supervisionada como a lei do modelo perceptron, a backpropagation entre outras. A figura 2.9 ilustra o procedimento do aprendizado supervisionado.



**Figura 2.9** Aprendizado supervisionado

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 podem variar em função do tipo de arquitetura utilizada

e da lei de aprendizagem. Alguns métodos de aprendizado não-supervisionado são Mapa Auto-Organizável de Kohonen, Redes de Hopfield e Memória Associativa Bidirecional (FREEMAN & SKAPURA, 1991). A figura 2.10 ilustra o procedimento do aprendizado não-supervisionado.



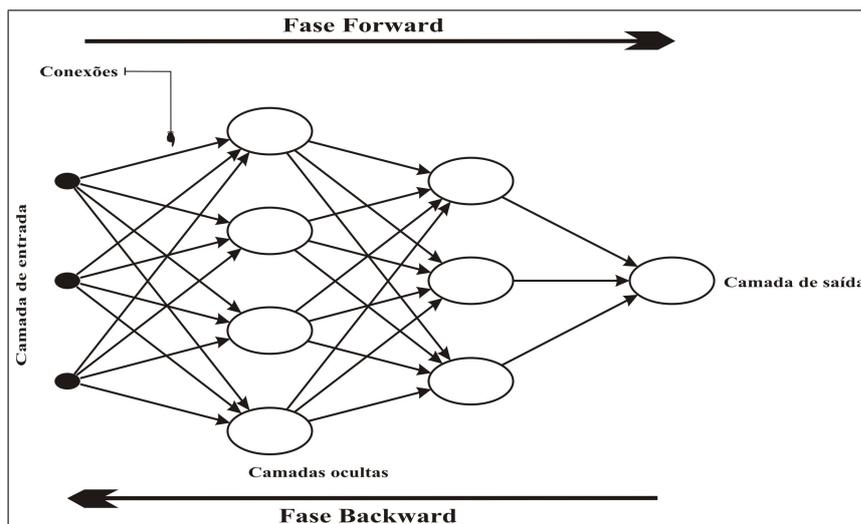
**Figura 2.10** Aprendizado não-supervisionado

Para aprendizado supervisionado, um algoritmo bastante usado em rede MLP é o backpropagation. Por isso, na subseção 2.2.4 seguinte o espaço será reservado para um mais detalhes sobre esse algoritmo.

#### 2.2.4 Backpropagation.

O algoritmo de aprendizado mais conhecido para o treinamento de redes de múltiplas camadas é o denominado backpropagation. São utilizados vetores de entrada associados aos seus respectivos vetores de saída desejada, até que a rede aproxime uma determinada função e possa, após esse processo, oferecer saídas adequadas a vetores de entrada diferentes daqueles com os quais foi treinada. O treinamento é feito em duas fases (fase forward e fase backward) para cada padrão apresentado, como mostra a figura 2.11.

O algoritmo backpropagation também é conhecido como regra delta generalizado, por ser baseado na regra delta proposta por Widrow & Hoff (1960) para a rede Adaline. Segundo Braga et al. (2000), as fases do algoritmo podem ser descritos da seguinte maneira: Os pesos e parâmetros são inicializados, isso é repetido até o erro ser mínimo ou até a realização de um dado número de ciclos. Para cada padrão de treinamento  $X$  a saída da rede é definida através da fase forward. São comparadas as saídas que foram produzidas com as saídas desejadas e os pesos dos nodos são atualizados através da fase backward.



**Figura 2.11** Fases do algoritmo backpropagation

### Algoritmo Backpropagation.

De acordo com Haykin (2001) os passos do algoritmo backpropagation são apresentados a seguir:

1. Iniciar os pesos: assumindo que nenhuma informação prévia está disponível, os pesos sinápticos e limiares são iniciados com números aleatórios que seguem uma distribuição uniforme;
2. Apresentação dos exemplos de treinamento: apresenta-se uma época de exemplos de treinamento à rede. Para cada exemplo do conjunto de treinamento executa-se os passos 3 e 4 apresentados a seguir:
3. Propagação: sendo um exemplo do conjunto de treinamento representado por  $(x(n), d(n))$ , com um vetor de entrada  $x(n)$  aplicado a camada de entrada de nós sensoriais e o vetor resposta desejada  $d(n)$  apresentado a camada de saída de nós computacionais, calcula-se os campos locais induzidos e os sinais funcionais, camada por camada da rede. O campo local induzido  $v_j^{(l)}(n)$  para o neurônio  $j$  na camada  $l$  é dado por:

$$v_j^{(l)}(n) = \sum_{i=0}^{m_0} \theta_{ji}^{(l)} y_j^{(l-1)}(n) \quad (2.6)$$

em que  $y_j^{(l-1)}(n)$  é o sinal de saída do neurônio  $i$  na camada anterior  $l - 1$ , na iteração  $n$ , e  $\theta_{ji}^{(l)}(n)$  é o peso sináptico do neurônio  $j$  da camada  $l$ , que é alimentado pelo neurônio  $i$  da camada  $(l - 1)$ . Para  $i = 0$ , temos  $y_0^{(l-1)}(n) = +1$  e  $y_{j0}^{(l)}(n) = b_{j0}^{(l)}(n)$  é o viés aplicado ao neurônio  $j$  na camada  $l$ . O sinal de saída do neurônio  $j$  na camada  $l$  é:

$$y_j^{(l)}(n) = \varphi(v_j(n)) \quad (2.7)$$

Se o neurônio  $j$  está na primeira camada oculta (i.e.,  $l = 1$ ), faz-se  $y_j^{(0)}(n) = x_j$ ,  $x_j$  é o  $j$ -ésimo elemento do vetor de entrada  $x(n)$ . Se o neurônio  $j$  na camada de saída (i.e.,  $l = L$  em que  $L$  é denominado a profundidade da rede), se faz  $y_j^{(L)} = 0_j(n)$ .

Calcula-se o sinal do erro

$$e_j^{(n)} = d_j^{(n)} - 0_j^{(n)} \quad (2.8)$$

onde  $d_{(n)}^j$  é o  $j$ -ésimo elemento do vetor resposta desejada  $d(n)$ .

4. Retropagação: Calcula-se os gradientes locais da rede definidos por:

$$\delta_j^{(n)} = \begin{cases} e_j^{(L)}(n) \varphi_j^{(L)}(v_j^{(L)}(n)) & \text{para o neurônio } j \text{ na camada de saída } L, \\ \varphi_j^{(l)}(v_j^{(L)}(n)) \sum_k \delta_k^{(l+1)}(n) \theta_{kj}^{(l+1)}(n) & \text{para o neurônio } j \text{ na camada oculta } l. \end{cases} \quad (2.9)$$

onde o apóstrofo  $\varphi_j^{(l)}(\cdot)$  representa a diferenciação em relação ao argumento. Assim, ajusta-se os pesos sinápticos da rede na camada  $l$  de acordo com a seguinte regra, conhecida como regra delta generalizada.

$$\theta_{ji}^{(n+1)} = \theta_{ji}^{(n)} + \alpha[\theta_{ji}^{(n)} - \theta_{ji}^{(n-1)}] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \quad (2.10)$$

onde  $\eta$  é a taxa de aprendizagem e  $\alpha$  é a constante de momento.

5. Iteração: repete-se os passos 3 e 4, apresentando a rede novos exemplos de treinamento até que o critério de parada seja satisfeito.

### **Algoritmo de treinamento rápido Levenberg-Marquardt.**

Para um treinamento mais rápido podem-se utilizar diversos artifícios, entre eles pode-se destacar a taxa de aprendizado variável e os algoritmos de gradiente conjugado. Existem ainda como alternativa os chamados de métodos de Newton para treinamento acelerado.

A diferença fundamental utilizada nos métodos de Newton em comparação ao algoritmo padrão está na matriz Hessiana (segunda derivada) do índice de erro aos valores atuais de pesos e bias. Os métodos de Newton sempre convergem mais rápido que os métodos de gradiente conjugado, mas, infelizmente, o cálculo da derivada segunda é complexo e demorado para determinar a matriz Hessiana. Assim, utiliza-se uma segunda classe de algoritmos que, apesar de baseados no método de Newton, não necessitam do cálculo da derivada segunda, são chamados de métodos Quase-Newton ou ainda de métodos secantes. Para mais detalhes sobre os algoritmos de otimização não linear consultar Bazaraa et al. (1993).

Semelhante aos algoritmos Quase-Newton, o algoritmo Levenberg-Marquardt foi desenvolvido para aproximar velocidade de treinamento de segunda ordem sem ter que calcular a matriz Hessiana. Quando a função erro faz a somatória quadrática, a matriz Hessiana pode ser aproximada como:

$$H = J^T J \quad (2.11)$$

E o gradiente pode ser calculado como:

$$g = J^T e \quad (2.12)$$

Onde  $J$  é a matriz Jacobiana que contém a primeira derivada dos erros da rede com os respectivos pesos e bias e  $e$  é o vetor de erros da rede. A matriz Jacobiana pode ser calculada através da técnica backpropagation padrão.

O algoritmo Levenberg-Marquardt usa esta aproximação Quase-Newton para a matriz Hessiana com a seguinte atualização:

$$x_{k+a} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (2.13)$$

Quando o módulo de  $\mu$  é zero, este é o método de Newton e usa a aproximação da matriz Hessiana. Quando  $\mu$  é grande, isto faz com que o gradiente descendente tenha um pequeno incremento a cada passo. Os algoritmos que usam os métodos de Newton são mais rápidos e mais precisos quanto menor for o erro mínimo desejado no treinamento.

Assim,  $\mu$  é diminuído depois de cada passo bem sucedido (redução do erro) e só é aumentado quando o erro aumentar. Deste modo, o erro sempre diminuirá a cada iteração do algoritmo. Este algoritmo aparece na bibliografia como sendo o método mais rápido para o treinamento de redes neurais de tamanho moderado. Além disso, encontra-se implementado de forma bastante eficiente como uma ferramenta do programa MATLAB sob o nome `trainlm` (BEALE ET AL., 2010).

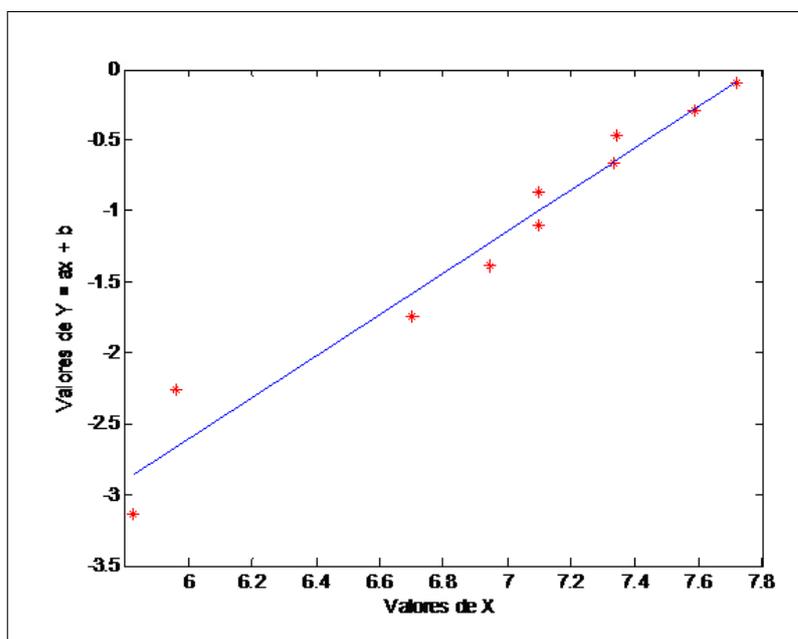
Outra metodologia sugerida para resolver o problema de modelagem de sistemas dinâmicos, mais especificamente o motor cc será apresentada na seção a seguir. Trata-se do método dos mínimos quadrados.

### 2.3 O Método dos Mínimos Quadrados.

O Método dos Mínimos Quadrados é uma importante ferramenta para solucionar problemas de ajuste de curvas. Seus conceitos são bastante difundidos em várias áreas da comunidade científica. De acordo com Aguirre (2004) a origem da idéia básica dele vem desde a época de Carl Friedrich Gauss (1777-1855), onde o mesmo desenvolveu o método e o utilizava nos trabalhos no âmbito da astronomia. Porém, só

anos mais tarde que foi chamado de Mínimos Quadrados por Adrien-Marie Legendre (1752-1833).

Observemos a figura 2.12, suponha que cada ponto vermelho seja um conjunto de várias grandezas de  $x$  e  $y$ , obtendo-se assim  $(x_1y_1), (x_2y_2), (x_3y_3), \dots, (x_ny_n)$ . Onde  $x$  é a variável controlada e  $y$  é a resposta, ou seja, a variável aleatória que se relaciona a  $x$  na equação. Então, para que haja uma relação linear é preciso que  $y = ax + b$ . O conceito de erro é tal que  $e = y - (ax + b)$ , ou seja, a diferença entre o valor esperado pelos valores medidos. Suponha-se que esse erro seja normalmente distribuído.



**Figura 2.12** Dispersão dos dados e a reta de ajuste pelos Mínimos Quadrados

De maneira formal, o Método dos Mínimos Quadrados trata de minimizar a soma dos quadrados dos resíduos (ou das diferenças).

$$\sum_i^n \varepsilon_i^2 = \sum_i^n (y_i - (ax_i + b))^2 \quad (2.14)$$

Segundo Meister (2006) a equação na forma matricial que relaciona as medidas aos parâmetros é formulada como:

$$\vec{y} = H \cdot \vec{x} \quad (2.15)$$

Onde  $\vec{y}$  é o vetor com  $m$  medidas,  $\vec{x}$  equivale ao vetor de  $n$  parâmetros que precisam ser estimados e  $H$  é a matriz  $m \times n$  que relaciona as medidas aos parâmetros. De maneira mais explícita, a equação (2.15) é melhor visualizada em (2.16).

$$\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{pmatrix} = \begin{pmatrix} t_1 & 1 \\ t_2 & 1 \\ \dots & \dots \\ t_m & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} \quad (2.16)$$

Quando  $n \neq m$ , a melhor solução corresponde ao vetor  $\vec{x}$  que aproxima ao máximo  $H\vec{x}$  do vetor  $\vec{y}$ . Em outras palavras, o vetor resíduo  $\vec{\varepsilon}(x)$  deve ser minimizado de acordo com a Equação (2.17) (MEISTER, 2006).

$$\vec{\varepsilon}(x) = (\vec{y} - H\vec{x})^t \cdot (\vec{y} - H\vec{x}) \quad (2.17)$$

Ou na notação de norma vetorial:

$$\vec{\varepsilon}(x) = \|\vec{y} - H\vec{x}\|^2 \quad (2.18)$$

A justificativa do uso da norma euclidiana é feita na medida em que, por definição, o quadrado da norma euclidiana de um vetor corresponde à soma dos quadrados dos componentes deste vetor. Segundo Datta (1995) a solução que apresenta a mínima norma euclidiana é chamada de solução de comprimento mínimo ou solução de norma mínima.

É importante notar que se minimizarmos os resíduos diretamente, um grande resíduo negativo pode ser anulado por um grande resíduo positivo, enquanto que

com o quadrado minimizamos os módulos das diferenças. Isso justifica o motivo de se usar esse critério ao invés de simplesmente minimizar apenas os resíduos ou o cubo das diferenças.

Segundo Souza (2003b), utilizando-se do cálculo diferencial, pode-se encontrar os melhores valores  $a$  e  $b$  da equação (2.16). Para isso, é preciso encontrar os valores nulos (condição de mínimo) das derivadas parciais de  $\bar{\epsilon}(x)$  com respeito a cada um de seus parâmetros respectivamente, veja a seguir:

$$\frac{\partial \epsilon}{\partial a} = -2 \sum_{i=1}^n (y_i - ax_i - b) = 0 \quad (2.19)$$

$$\frac{\partial \epsilon}{\partial b} = -2 \sum_{i=1}^n x_i (y_i - ax_i - b) = 0 \quad (2.20)$$

Sabendo que:

$$\begin{aligned} \sum_{i=1}^n (y_i - ax_i - b) &= \sum_{i=1}^n y_i - \sum_{i=1}^n ax_i - \sum_{i=1}^n b \\ &= \sum_{i=1}^n y_i - \left( \sum_{i=1}^n x_i \right) a - nb \end{aligned} \quad (2.21)$$

e ainda que:

$$\sum_{i=1}^n x_i (y_i - ax_i - b) = \sum_{i=1}^n x_i y_i - \left( \sum_{i=1}^n x_i \right) b - \left( \sum_{i=1}^n x_i^2 \right) a \quad (2.22)$$

É possível chegar ao sistema de equações, conhecido como “equações normais” do problema. As incógnitas encontradas são os parâmetros  $a$  e  $b$  que se deseja estimar.

$$\begin{cases} nb + (\sum_{i=1}^n x_i) a = \sum_{i=1}^n y_i \\ (\sum_{i=1}^n x_i) b + (\sum_{i=1}^n x_i^2) a = \sum_{i=1}^n x_i y_i \end{cases} \quad (2.23)$$

A próxima subseção o espaço é reservado para a qualidade de ajuste do modelo, ou seja, através coeficiente de determinação, parâmetro  $r^2$ , é possível observar se com os dados apresentados conseguiu-se obter um bom modelo pelo método dos Mínimos Quadrados.

### 2.3.1 Qualidade de ajuste.

O teste de qualidade de ajuste visa investigar o quanto a equação estimada se ajusta aos pontos  $(x, y)$  dados. Esse teste é feito através da definição do coeficiente de determinação  $r^2$ , onde o mesmo assume valores entre 0 e 1. Segundo Datta (1995), quanto mais próximo o valor desse coeficiente se aproximar da unidade melhor é a relação linear do modelo aos dados coletados. E ainda, se um modelo tiver um valor acima de 0,9 pode-se considerar que o modelo se ajusta bem aos dados. Caso contrário, ou seja, para  $r^2$  pequeno conclui-se que o modelo não se ajusta bem aos dados ou que provavelmente não há uma relação linear entre  $x$  e  $y$ .

A expressão para determinação do coeficiente de determinação  $r^2$  é dada por (SOUZA, 2003b):

$$r^2 = 1 - \frac{\sum_{i=1}^n (y_i - ax_i - b)^2}{\sum_{i=1}^n y^2 - \frac{1}{n} (\sum_{i=1}^n y_i)^2} \quad (2.24)$$

A seguir uma breve revisão bibliográfica foi realizada visando identificar a evolução das pesquisas no que se refere à modelagem de sistemas dinâmicos pelos métodos aqui sugeridos.

## Capítulo 3

### ESTADO DA ARTE

A fim de buscar referências pertinentes ao problema de pesquisa dessa dissertação foi realizada uma revisão bibliográfica. Esta investigação tem a finalidade de conhecer a evolução das pesquisas sobre RNA's e o método de Mínimos Quadrados nos últimos anos principalmente quando aplicados em modelagem de sistema dinâmicos. A seguir são apresentados alguns trabalhos usando esses métodos com aplicação semelhante ao proposto neste trabalho.

Vargas (1997) fez investigações a respeito de algoritmos de treinamentos que obtivessem 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). A que mais foi focada em seus estudos foram as Redes Neurais Recorrentes com a implementação dos algoritmos desenvolvidos por ele. 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 identificar outros possíveis métodos a serem testados para a modelagem do problema de seu trabalho, sem sair do foco de RNA. Primeiramente ele optou por implementar técnicas mais comuns e mais difundidas. Em seguida, foi buscado a solução através de técnicas mais avançadas.

Oliveira (1998) propôs a técnica de estimação de parâmetros de um motor de indução por meio do método dos Mínimos Quadrados Recursivo. Os procedimentos apresentados neste trabalho foram comprovados através de dados de simulação e experimentais. Os resultados confirmam o bom desempenho do método proposto. Outro fator a ser observado é que a estimação de forma on-line não foi possível de ser conseguido. Ou seja, estimar os parâmetros no mesmo instante do funcionamento do motor. Com a estimação on-line é possível implementar sistemas de controle adaptativo, para isso, basta usar os parâmetros estimados dinamicamente e ajustar os

ganhos do controlador. O autor deixa este problema como uma das sugestões para trabalhos futuros.

Assim como os motores cc, os motores de indução têm bastante utilidade, principalmente quando se trata de aplicações industriais. Por isso, a necessidade de estimação dos seus parâmetros e definição do modelo mais fiel possível ao sistema. O trabalho feito por Oliveira (1998) tem por objetivo aplicar e verificar a eficiência da técnica de estimação pelos mínimos quadrados recursivos. O problema proposto nesta dissertação tem a finalidade de emular duas técnicas de estimação para possível controle.

Tinós (1999) desenvolveu um sistema de detecção e diagnóstico de falhas para um robô manipulador. A sua motivação de utilizar RNA para o problema de pesquisa foi a baixa complexidade, se comparado a modelagem analítica. Essa “simplicidade” evita que erros de modelagem possam ocultar os efeitos das falhas e ainda serem fontes de alarmes falsos. O modelo de RNA desenvolvido foi o MLP treinado com o algoritmo por retropropagação do erro e também um modelo RBF com quatro tipos diferentes de algoritmos de treinamentos, dentre eles o Mapa Auto-Organizável de Kohonen. O resultado foi bastante promissor, até mesmo para conjuntos de padrões não treinados.

É importante para este trabalho investigar a evolução de estudos sobre manipuladores robóticos principalmente ao usar RNA. Tendo em vista que os motores cc têm grande aplicabilidade como atuador destes equipamentos. Com isso, o enriquecimento teórico a respeito dessa aplicação se expande. Tinós (1999) fez relevantes comentários de como fazer com que o algoritmo tenha uma convergência rápida.

Cerqueira (2001) usa um modelo de manipulador robótico para testar a eficiência e fazer a análise de convergência do identificador de sistemas não lineares e multivariáveis. O autor optou por usar as RNA's do tipo MLP, onde os parâmetros eram ajustados pelo procedimento on-line. Ao final do trabalho, ele obteve bons resultados, e confirmou a qualidade de um identificador capaz de aproximar modelos entrada-saída multivariáveis e não lineares.

Quando se trata de RNA do tipo MLP é normal encontrarmos estimadores que fazem a modelagem apenas com os dados medidos, o que chamamos de treinamento

offline ou em batelada. Porém, quando o sistema é dinâmico, como é o caso do sistema que contém o motor cc, em alguns casos torna-se necessário a identificação do modelo com o equipamento em funcionamento, o que chamamos de estimadores online. O trabalho desenvolvido por Cerqueira (2001) serve como uma importante fonte de referência para este tipo de problema. Outras contribuições foram feitas para o algoritmo de treinamento backpropagation, um algoritmo de aprendizagem adaptativo entre outras.

O desenvolvimento de um sistema de controle neural pode ser abordado de duas maneiras: métodos diretos e métodos indiretos. Direto significa que o controlador é neural, sendo que os seus pesos sinápticos são ajustados diretamente com base nos parâmetros da planta. O método indireto necessita de um modelo neural da planta para ajustar os parâmetros do controlador, que também pode ou não ser baseado nas RNA's. Maitelli & Filho (2003) apresentam um sistema de controle indireto, porém híbrido. Neste trabalho, a RNA do tipo MLP com aprendizagem baseada em Error Back Propagation foi utilizada apenas para a identificação da resposta dinâmica não linear da planta e o controlador convencional usado para regular o funcionamento da planta.

Reforçando a idéia do problema de pesquisa proposto neste trabalho Maitelli & Filho (2003) Comentaram a relevância de uma boa identificação dos modelos e ainda faz uma aplicação para um sistema de controle. Com o modelo encontrado de forma online por meio de RNA, ele pode aplicar de forma eficiente a um sistema de controle. Este poderia ser também neural como poderia ser convencional, a segunda opção foi escolhida caracterizando-se como um controlador híbrido.

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. A comprovação da eficiência desse método vem também com a observação do custo computacional e de implementação que foram bem baixos.

Um motor de indução é um equipamento que possui uma dinâmica de funcionamento bem mais complexa, se comparado ao motor cc, pois existem ainda mais

parâmetros envolvidos em seu modelo matemático. O trabalho de Oliveira (2004) é uma importante referência para essa dissertação. Nele podemos encontrar um estudo rigoroso de aplicação das RNA e ainda o uso de um algoritmo de treinamento (Filtro de Kalman) não comumente usado, porém com um excelente desempenho.

Sabendo da importância que um transformador possui para sistemas elétricos de potência. Meister (2006), em sua dissertação de mestrado, aplicou o Método dos Mínimos Quadrados para fazer a identificação da função de transferência e os parâmetros de um transformador. Isso é feito para melhor entender o funcionamento deste equipamento e assim prever falhas ou então sugerir melhorias na fabricação. Conclui-se nesse trabalho que o modelo identificação teve ótima aproximação do sistema real, pois obtiveram pequenos erros nas medidas do sistema real com o modelo estimado.

O trabalho feito por Meister (2006) vai ao encontro com o trabalho proposto aqui, pois ambos têm a finalidade de encontrar um modelo para o sistema para assim poder fazer estudos sobre o equipamento sem usá-lo fisicamente. Porém, esta dissertação vai além, pois identifica o motor cc. Além disso, tem a intenção de comparar qual método representa melhor o modelo do motor cc.

Furtado (2008) propôs uma comparação entre técnicas de identificação de parâmetros 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 não linear, porém não associou esse problema a uma aplicação prática. A proposta dessa dissertação é usar um sistema também não linear, porém, um sistema real e ainda sugerindo suas aplicações.

Com a intenção de aplicar técnicas de Controle Clássico do tipo PID e verificar a mais adequada, Sousa (2009) intensificou seus estudos para controlar a velocidade

de um motor cc. Para fazer a identificação desse equipamento ele usou o Método dos Mínimos Quadrados Recursivo (MQR), que relaciona a velocidade do rotor com a tensão de armadura. O resultado dessa estimação foi positivo obtendo erro bastante pequeno.

O trabalho de Sousa (2009) usou uma variação do Método dos Mínimos Quadrados convencional, ou seja, MQR. Observa-se nesse trabalho que a finalidade maior era de encontrar qual estratégia de controle melhor serviria para controlar a velocidade do motor cc em questão. O MQR nesse caso serviu para estimar os parâmetros do sistema, o que diferencia desse projeto é o pouco aprofundamento e ênfase nas técnicas de estimação, principalmente emulando métodos estatísticos com inteligência computacional.

Moreira (2010) projetou e implementou um controlador de velocidade digital para um motor cc. Nesse trabalho, o mesmo usou o Método dos Mínimos Quadrados para estimar os parâmetros do motor cc continuamente. Com isso, o sistema de controle se caracterizava como um Controle Adaptativo, pois os parâmetros eram ajustados dinamicamente ao controlador. A vantagem de um sistema como esse é a possibilidade de usá-lo em outros motores CC sem precisar fazer a identificação dos parâmetros novamente, porém, o risco da não estabilidade do sistema de controle é inevitável.

No que se refere à questão do Método dos Mínimos Quadrados, o trabalho de Moreira (2010) vai ao encontro do problema aqui proposto. Mas o que difere é que o trabalho feito por ele resolve um problema de controle que utiliza esse método. Admitindo que o mesmo seja uma das melhores soluções a ser seguida para a identificação. Já no problema de pesquisa em questão a proposta é emular a eficiência do Método dos Mínimos Quadrados e RNA's para estimar os parâmetros do modelo.

Silva & Otal (2010) fizeram um estudo comparativo de dois métodos de modelagem aplicados a sistemas dinâmicos. Após os testes, eles puderam constatar que a RNA do tipo MLP, treinada pelo algoritmo backpropagation resultou em melhores resultados. O sistema dinâmico usado para o estudo foi uma coluna de destilação piloto e o segundo método usado para a estimação foram os modelos auto-regressivos. Os dados a respeito da modelagem por meio de modelos auto-regressivos foram obtidos do trabalho de Carvalho et al. (2004). Portanto, somente o modelo via RNA que foi

desenvolvido no TCC.

Uma fonte de informação útil para esta dissertação foi observar que em outros trabalhos a RNA já apresentara bom desempenho se comparado com algum outro método. Para este caso os modelos auto-regressivos apresentarem um desempenho inferior. Um momento importante de ser observado neste TCC foi o critério usado para a escolha de qual modelo de RNA usar, somente após diversos testes realizados com configurações de redes diferentes que Silva & Otal (2010) chegaram ao modelo final. Sistemas dinâmicos necessita de bastante cuidado na sua modelagem, pois pequenos erros podem causar imprecisão num sistema de controle baseado neste modelo.

No capítulo a seguir todos os estudos serão concentrados em investigar o equipamento utilizado para auxiliar na aplicação dos dois métodos de modelagem do motor cc.

## Capítulo 4

### KIT FEEDBACK® - SERVO FUNDAMENTALS TRAINER

Os módulos Feedback® são uma combinação de componentes eletromecânicos, circuitos eletrônicos e software utilizados para treinamento dos fundamentos de controle. O módulo utilizado neste trabalho chama-se Kit Feedback®- Servo Fundamentals Trainer. Ele é um ótimo equipamento para desenvolvimento de pesquisas e práticas de laboratórios, pois simula de maneira aproximada e eficiente um sistema real. Sua composição é dada 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. O kit fica situado no Laboratório de Automação Industrial (LAI) na Área do Departamento de Engenharia de Controle e Automação do Instituto Federal Fluminense (IFF).

Como elementos para a pesquisa serão utilizados um microcomputador (PC), para que sejam configuradas todas as estratégias desenvolvidas, além da aquisição dos dados. O software que será utilizado para programação do experimento é o MATLAB® com o Toolbox Real-Time Workshop, destinado para experimentos em tempo real. Será utilizada também a UM-33-100 e ainda a UD-33-120.

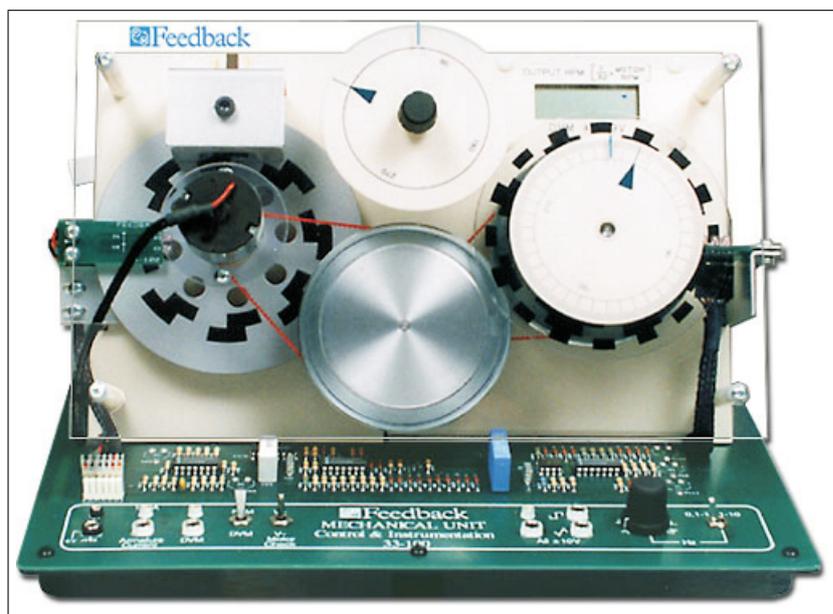
Na próxima seção uma apresentação da UM-33-100 será feita.

#### 4.1 Unidade Mecânica-33-100.

A UM-33-100 consiste em um conjunto que contém os componentes eletromecânicos do servossistema, montados na parte superior, e sua eletrônica de suporte. Os componentes eletromecânicos compreendem um motor de corrente contínua (cc), um tacogerador analógico, potenciômetros de entrada e saída analógicas, codificadores digitais absolutos e incrementais e freio magnético.

A eletrônica de suporte compreende: o amplificador de potência, um gerador de baixa frequência de ondas de forma senoidal, quadrada e triangular para testes, circuitos de leitura do codificador e display de LCD de indicação de velocidade. A UM-33-

100 é comum tanto para a unidade digital quanto para a analógica. Como elementos para este trabalho serão utilizados da UM-33-100, o motor cc e o sensor de velocidade analógico (tacogerador) acoplado ao eixo do motor. Para maiores detalhes da unidade mecânica deve-se consultar Feedback Instruments Ltda. (1995a). A unidade mecânica é apresentada na figura 4.1.



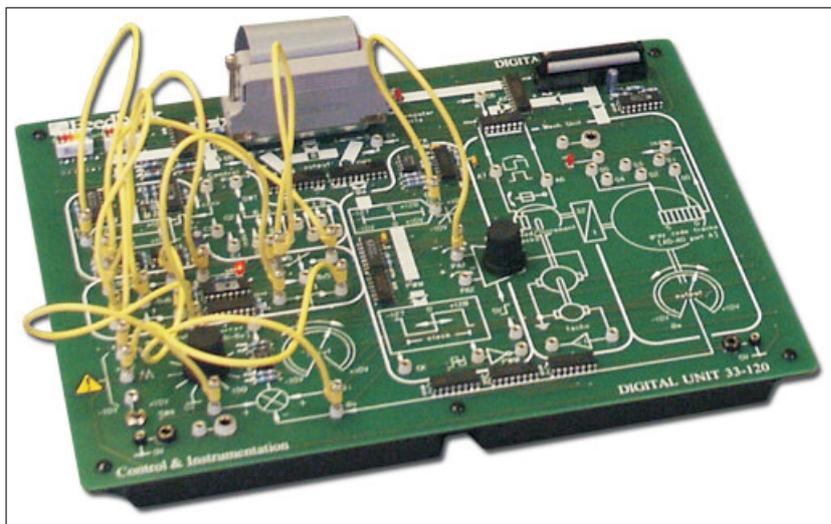
**Figura 4.1** UM - 33-100. (Fonte: Feedback Instruments Ltda. (1995b))

A UD-33-120 é apresentada na seção a seguir.

## 4.2 Unidade Digital-33-120.

A UD-33-120 consiste em uma placa com circuitos digitais que oferece uma interface digital entre a unidade mecânica e o PC. Ela contém os circuitos dos conversores analógico-digital (A/D) e digital-analógico (D/A) para a conversão dos sinais do kit para o PC e do PC para o kit respectivamente. Possui ainda um sistema de introdução de falhas para se fazer testes de robustez a falhas, comumente utilizados em estudos de sistemas de controle.

Como elementos para este trabalho serão utilizados da UD-33-120 os conversores A/D e D/A. Para maiores detalhes da UD-33-120 deve-se consultar os manuais da Feedback Instruments Ltda. (1995b). A unidade digital é apresentada na figura 4.2.



**Figura 4.2** UD - 33-120. (Fonte: Feedback Instruments Ltda. (1995b))

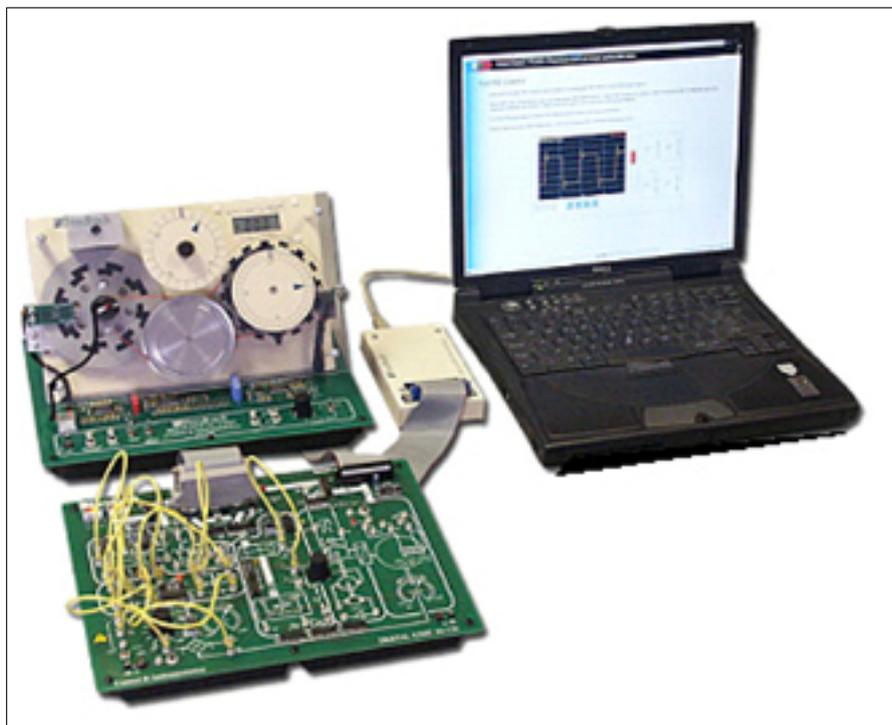
Todas as unidades do kit que foram utilizadas são conectadas ao PC. A comunicação segue um padrão que será apresentado na próxima seção.

### 4.3 Esquema de comunicação.

O kit é alimentado por uma fonte (+15V e -15V a 1,5A e +5V a 0,5A). Ela é conectada a UM-33-100 que se comunica com a UD-33-120 por um cabo de 34 vias. Este cabo fornece também a alimentação elétrica (+5V) proveniente da UM-33-100. A interligação entre a UD 33-120 e o PC é feita através de um cartão de interface (placa de aquisição) paralela MIC926, instalado dentro do PC e ligado a um slot extra de expansão deste. Este cartão utiliza um cabo flexível de 37 vias.

A figura 4.3 ilustra a unidade mecânica conectada à unidade digital e esta ao PC. O protocolo de comunicação utilizado é um protocolo proprietário especificado pelo manual da placa de aquisição. Este esquema de comunicação é ilustrado em forma de diagrama de blocos pela figura 4.4.

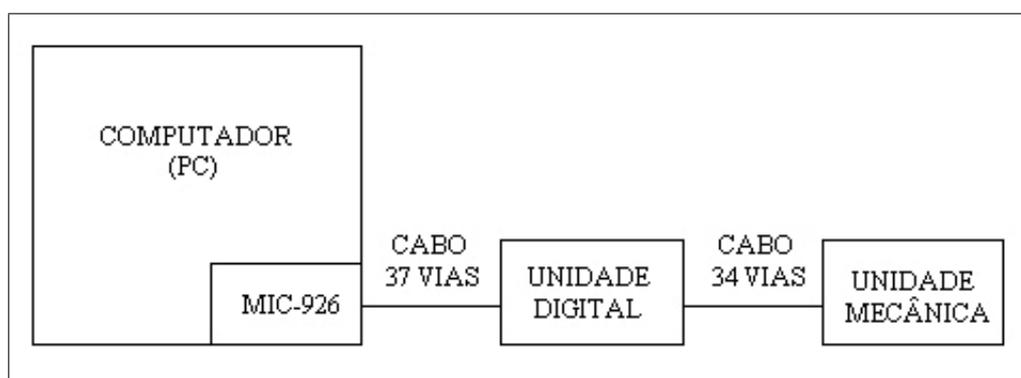
O motor cc é o componente do kit mais importante para o estudo, pois é nele que serão aplicados os métodos. Por conta dessa importância a seção seguinte fala sobre ele bem como apresenta todas as variáveis envolvidas e ainda o seu modelo matemático.



**Figura 4.3** Kit Feedback instalado. (Fonte: Feedback Instruments Ltda. (1995b))

#### 4.4 Motor cc.

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,



**Figura 4.4** Esquema de comunicação entre a UM-33-100, a UD-33-120 e o PC.

máquinas-ferramentas e atuadores de servoválvulas (NAVARRO & LEITE, 2010). Por isso, a identificação de um modelo adequado para um possível desenvolvimento de um sistema de controle deve ser obtida.

Segundo Navarro & Leite (2010) 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 se encontra entre -10V e +10V. Para maiores detalhes sobre o funcionamento de motores cc ver Siemens (2006). A figura 4.5 ilustra um motor cc.

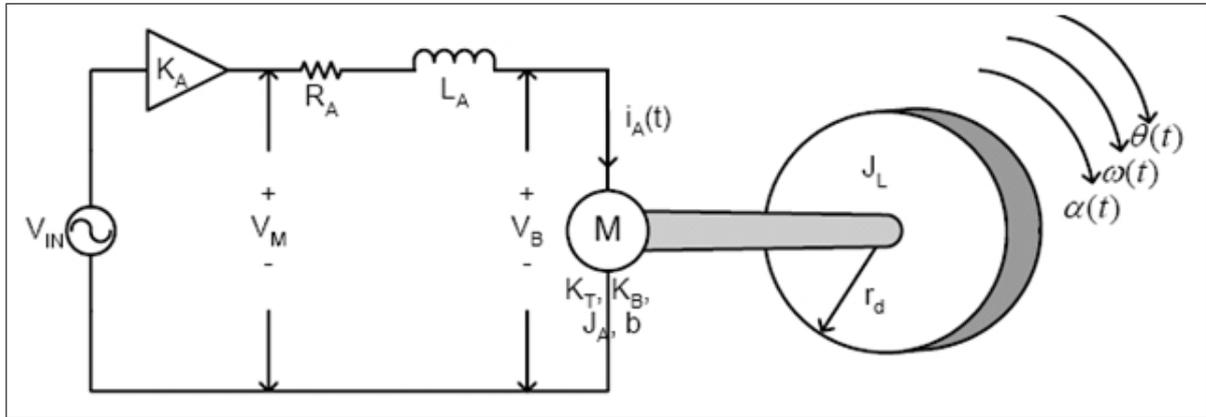


**Figura 4.5** Motor cc. (Fonte: Action Technology (2011))

O processo de obtenção de um modelo matemático de um motor cc comandado pela tensão aplicada nos seus terminais é bem conhecido. Diversas variáveis e parâmetros estão envolvidos no funcionamento do motor cc. Para uma melhor visualização das mesmas, é apresentada a figura 4.6.

A descrição das variáveis envolvidas está na tabela 4.1.

Segundo o procedimento realizado em Dorf & Bishop (2004), pela abordagem de controle clássico, a seguinte função de transferência pode ser obtida:



**Figura 4.6** Detalhamento das variáveis do motor cc em questão (Adaptado de Dorf & Bishop (2004)).

$$\frac{\Omega(s)}{V_{IN}(s)} = \frac{K_A K_T}{L_A J_T s^2 + (R_A J_T + L_A b) s + (R_A b + K_B K_T)} \quad (4.1)$$

Note que esta função de transferência é de segunda ordem. Como uma das desvantagens de se utilizar o método dos mínimos quadrados se refere exatamente à esse fator. Em vista disso, a função de transferência do motor cc que será utilizada é uma aproximação linear do motor real, onde os efeitos de segunda ordem serão desprezados. Esta aproximação é bastante freqüente em motores cc, mais informações podem ser encontradas em Dorf & Bishop (2004) e Leite et al. (2006). O modelo matemático simplificado da equação (4.1) do motor cc é apresentado na equação (4.2). E sua representação em diagrama de blocos é ilustrada através da figura 4.7.

$$\frac{\Omega(s)}{V_{IN}(s)} = \frac{K_m}{T_m s + 1} \quad (4.2)$$

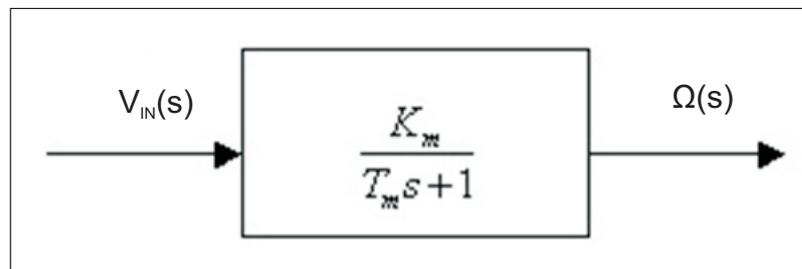
Onde:  $K_m$  representa o ganho de resposta do motor cc e  $T_m$  a constante de tempo do mesmo.

Observe que as funções de transferências estão sendo apresentadas no domínio de Laplace. Para isso, foi necessário fazer a transformada de Laplace da equação diferencial do modelo.

A transformada de Laplace é um método simples de transformar um problema

**Tabela 4.1** Variáveis envolvidas em um motor cc

Símbolo	Descrição
$V_{IN}(t)$	Tensão de comando do motor (Volts)
$K_A$	Ganho do amplificador (Volts/Volts)
$V_M(t)$	Tensão nos terminais do motor (Volts)
$R_A$	Resistência da armadura do motor (Ohms)
$L_A$	Indutância da armadura do motor (Henry)
$V_B(t)$	Força contra eletromotriz do motor (Volts)
$i_A(t)$	Corrente na armadura do motor (Ampères)
$K_T$	Constante de torque do motor ( $Nm/A$ )
$K_B$	Constante de velocidade de força contra eletromotriz ( $Volt.Sec/rad$ )
$J_A$	Inércia da armadura do motor ( $Kg.m^2$ )
$b$	Coefficiente de amortecimento viscoso ( $Kg.m^2/Sec$ )
$J_L$	Inércia da carga ( $Kg.m^2$ )
$r_d$	Raio do disco (m)
$\alpha(t)$	Aceleração angular do motor ( $rad/s^2$ )
$\omega(t)$	Velocidade angular do motor ( $rad/s$ )
$\theta(t)$	Posição angular do motor (rad)
$T(t)$	Torque do motor (Nm)

**Figura 4.7** Diagrama de blocos da função de transferência simplificada do motor cc

com valores iniciais, em uma equação algébrica, de modo a obter uma solução deste problema de uma forma indireta, sem o cálculo de integrais e derivadas para obter a solução geral da equação diferencial. Segundo Ogata (2004) o método da transformada de Laplace é um método operacional que pode ser usado com vantagens para resolver equações diferenciais lineares. Usando transformadas de Laplace, podem-se converter muitas funções comuns, tais como funções senoidais, senoidais amortecidas, exponenciais em funções algébricas de uma variável complexa  $s$ . A solução da equação diferencial pode então ser achada através do uso de uma tabela de transformadas de Laplace ou pelo uso da técnica de expansão em frações parciais.

Foi brevemente comentado que o kit possui três tipos de sensores de velocidade do motor, são dois codificadores (um incremental e outro absoluto) e ainda um tacogerador. Na próxima seção serão comentados mais detalhes sobre o sensor utilizado.

#### 4.5 Sensor de velocidade (tacogerador).

O sensor de velocidade utilizado é um tacogerador acoplado ao eixo do motor cc da UM-33-100. Ele é um gerador que transforma a energia mecânica de acionamento do mesmo em energia elétrica na forma de tensão contínua induzida em seus terminais. Essa tensão gerada é proporcional à velocidade angular a qual o tacogerador é submetido, sendo esse sinal analógico de tensão (-5V +5V) utilizado como medição da velocidade do disco (Planta). O ganho do sensor é:  $2,5V/1000$  RPM. O modelo matemático do tacogerador é o próprio ganho, como demonstrado na equação (4.5), que representa a sua função de transferência. E na figura 4.8 ela é ilustrada na forma de diagrama de blocos.

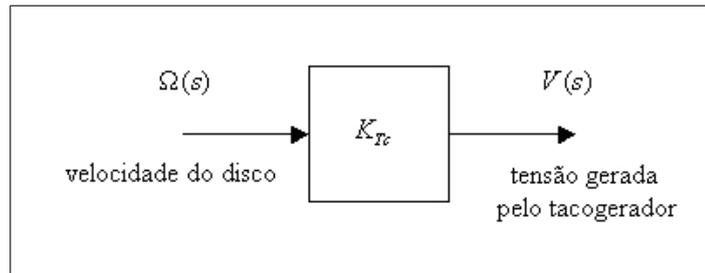
$$v(t) = K_{Tc} \cdot \omega(t) \quad (4.3)$$

Aplicando-se a transformada de Laplace e considerando-se as condições iniciais nulas, temos:

$$V(s) = K_{Tc} \cdot \Omega(s) \quad (4.4)$$

$$\frac{V(s)}{\Omega(s)} = K_{Tc} \quad (4.5)$$

Na próxima seção será apresentado mais detalhes sobre os conversores A/D.



**Figura 4.8** Diagrama de blocos da função de transferência do tacogerador.

#### 4.6 Conversor Analógico-Digital (A/D).

O conversor A/D é um dispositivo eletrônico capaz de gerar uma representação digital de uma grandeza analógica. Como o sensor do kit produz um sinal analógico de saída (-5V +5V), tensão esta que é contínua, é necessária a conversão desse sinal para digital (palavra de 8 bits), que é feita no conversor A/D da UD-33-120, para esse sinal ser transferido para o computador, onde será processado. A resolução de um conversor A/D é dada pelo valor máximo analógico dividido pelo valor máximo de combinações de bits. Com isso, entende-se que a resolução é o menor valor que pode ser representado por um conversor A/D. Desta forma qualquer valor que esteja abaixo disso não provocará nenhuma mudança na saída do conversor.

Na próxima seção será comentado um pouco mais sobre os conversores que D/A.

#### 4.7 Conversor Digital-Analógico (D/A).

A função básica do conversor D/A é converter a representação digital (palavra) em uma tensão analógica equivalente. O sinal digital proveniente do computador deve ser convertido em sinal analógico (-10 V +10 V) para ser aplicado no atuador (motor cc), o que é feito pelo conversor D/A da UD-33-120. A representação matemática do conversor D/A é mostrada pela equação (4.6).

$$V_0 = V_{fs}[A_1 2^{-1} + A_2 2^{-2} + \dots + A_n 2^{-n}] \quad (4.6)$$

Onde  $V_{fs}$  representa uma tensão de referência a qual determina a escala completa “range” de tensão de saída do conversor, e  $A_1$  a  $A_n$  representam os dígitos binários ou bits da palavra de entrada.  $A_1$  é chamado de bit mais significativo (MSB) e corresponde à tensão de  $V_{fs}/2$ .  $A_n$  é o bit menos significativo (LSB) e corresponde a  $V_{fs}/2^n$ . A resolução do conversor é a menor variação analógica que pode ser produzida pelo conversor e é igual ao valor de LSB em volts, entretanto, sempre é especificada como porcentagem da escala ou ainda como resolução n-bit.

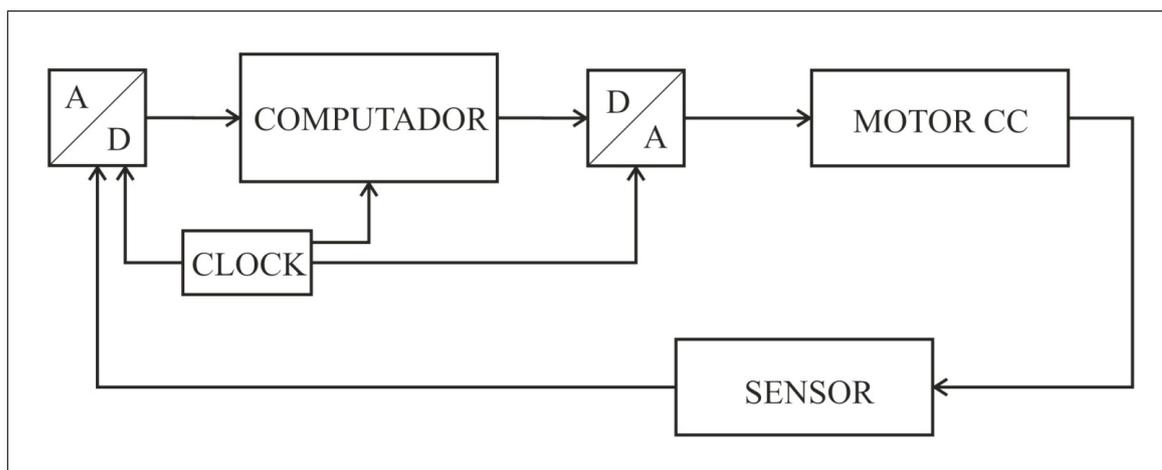
Após o conhecimento do kit a ser utilizado e toda a teoria envolta dos métodos de estimação, para fazer a modelagem é preciso estabelecer uma metodologia. Então, o próximo capítulo esclarece de forma mais clara todos os passos do processo de estimação.

## Capítulo 5

### METODOLOGIA E IMPLEMENTAÇÃO

#### 5.1 Funcionamento do kit.

O Sistema para a estimação dos parâmetros do motor cc é composto pelo atuador que é o próprio motor, pelo sensor (tacogerador) que está diretamente acoplado ao eixo do motor (estes elementos pertencem a UM-33-100), pelos conversores A/D e D/A (que pertencem a UD-33-120) e pelo computador que é um PC onde será configurada a lógica digital com a utilização de um software. Neste caso optou-se por utilizar o MATLAB<sup>®</sup>, que é um software bastante conhecido no meio de pesquisa e não é limitado apenas ao equipamento. A vantagem desta característica é que, ao contrário do software específico do kit, o MATLAB<sup>®</sup> não é limitado apenas a este equipamento, o que possibilita utilizá-lo em outros motores que não fazem parte do kit. A figura 5.1 mostra uma visão geral do funcionamento do sistema para a estimação dos parâmetros do motor cc.



**Figura 5.1** Visão geral do sistema em diagrama de blocos.

Acompanhando a figura 5.1 e considerando-se o estado inicial de repouso de todo o sistema, ao entrar em funcionamento, o motor cc começará a produzir um torque no seu eixo, conseqüentemente o disco irá adquirir uma velocidade angular  $\omega(t)$ .

Esta velocidade é diretamente proporcional ao torque produzido. A velocidade será medida pelo sensor, um tacogerador, que produzirá uma tensão contínua induzida, diretamente proporcional à velocidade do eixo do motor. Esse sinal analógico do sensor é então convertido para um sinal digital de 8 bits, através do conversor A/D, e enviado ao PC. Através do MATLAB<sup>®</sup>, o PC armazena os dados vindos do conversor A/D e ainda envia comandos para o motor cc por meio de sinal digital. Este sinal digital de saída será então convertido para um sinal analógico, através do conversor D/A, e aplicado ao motor cc para regular a sua velocidade. O clock é um gerador de pulsos, comum para os conversores A/D e D/A. Estes pulsos representam a taxa de amostragem, pois um sistema digital não funciona em tempo contínuo.

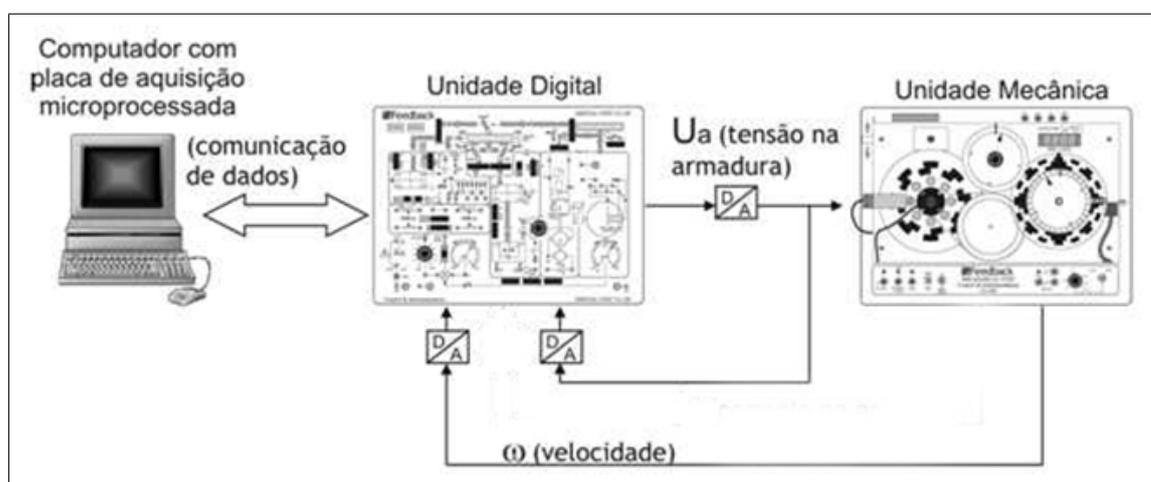
A constante de tempo da planta é de aproximadamente 0,21s (aproximadamente 5 Hz). Portanto, de acordo com o Teorema da Amostragem de Shannon-Nyquist descrito em Franklin et al. (1990), a mínima frequência de amostragem admissível é o dobro da maior frequência da planta. Isto significa que a frequência necessária é 10 Hz. Mas ainda de acordo com Franklin et al. (1990) uma escolha razoável pelo sistema ter banda espectral ilimitada, seria de 6 a 10 vezes mais esta frequência de 5 Hz. Isto resulta em 50 Hz no máximo. Como o conversor A/D fornecido pela unidade digital tem frequência de amostragem igual a 100 Hz no máximo, este valor foi escolhido sabendo das implicações do referido teorema. Outro aspecto importante é o tempo gasto pelo computador para realizar os cálculos, o período escolhido para os integradores foi de 0,001s. Isto significa que entre um período de amostragem do conversor A/D, o computador de controle realiza 10 passos de integração.

A coleta dos dados se faz necessária para os dois métodos. Se não for feito um adequado levantamento é possível que os modelos estimados por RNA ou pelos mínimos Quadrados não reflitam fielmente a realidade. Sendo necessária uma coleta mais eficaz. A próxima seção explica detalhadamente o procedimento para se fazer este levantamento.

## 5.2 Coleta de dados.

Definiu-se anteriormente que o problema em questão é o de estimar os parâmetros de um motor cc pelo Método dos Mínimos Quadrados e no padrão de RNA que se

comporte de maneira mais próxima do sistema real. Em seguida, comparar qual dos dois métodos apresentam um melhor desempenho. 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. É necessário ser feito nesse momento um levantamento prévio do comportamento dinâmico do sistema em questão. Segundo Ogata (2004) freqüentemente esse levantamento é feito analisando a resposta transitória a uma excitação em degrau, pois este sinal é fácil de ser gerado e corresponde a uma solicitação suficientemente severa. Este tipo de levantamento nada mais é que uma variação brusca na entrada, para verificar o seu comportamento na saída. Para este trabalho também optou-se por utilizar esse tipo de excitação. A figura 5.2 ilustra a forma de coleta dos dados.



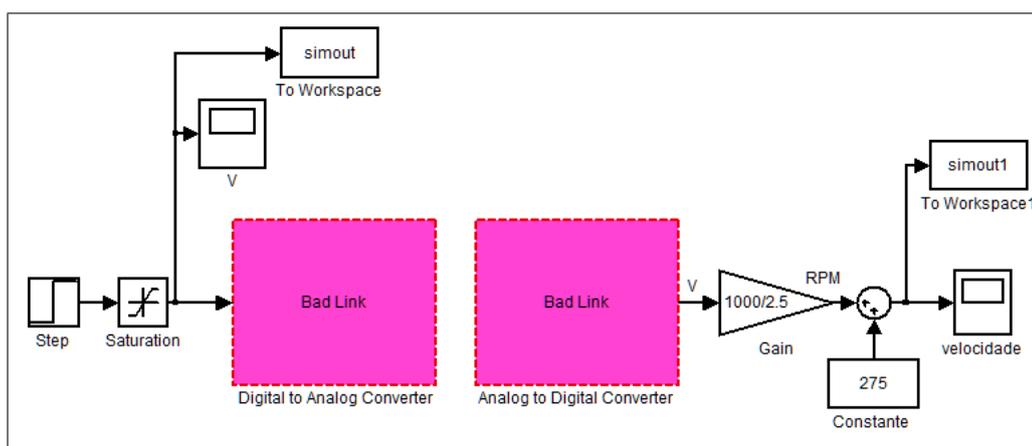
**Figura 5.2** Ilustração do esquema de coleta de dados (Fonte: Leite et al. (2006))

Observando a figura 5.2, após enviar o comando de tensão para o funcionamento do motor cc, o computador registra numa matriz o valor de tensão enviado e a resposta da velocidade do atuador. 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. Conforme já apresentado, o motor cc utilizado trabalha numa faixa de -10V até 10V. 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. Esses valores coletados foram armazenados no MATLAB® na forma de uma matriz  $A_{m \times n}$  (onde  $m = 1000$  e  $n = 2$ ). A tabela 5.1 resume os detalhes do experimento para a coleta de dados.

**Tabela 5.1** Detalhes técnicos da coleta dos dados (adaptado de Leite (2007))

<b>Período de amostragem</b>	0,01s
<b>Resolução do conversor D/A</b>	8 bits
<b>Resolução do conversor A/D</b>	8 bits
<b>Software utilizado</b>	Matlab®/ Real-time Workshop
<b>Target</b>	c/c++ for feedback experiments
<b>Sensor de velocidade</b>	Tacogerador cc
<b>Faixa de tensões aplicáveis ao motor</b>	-10v a 10v
<b>Tempo das simulações</b>	10s
<b>Tipo de entrada</b>	Degrau
<b>Amplitudes</b>	-10v até 10v
<b>Instante de aplicação do degrau</b>	5s

No software de simulação Simulink® foi montado um diagrama de blocos para que sejam feitos os experimentos de coleta dos dados. Esse diagrama corresponde a figura 5.3.

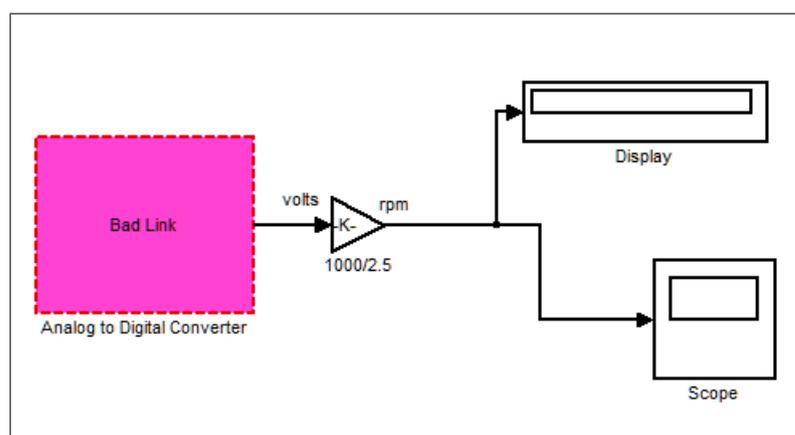
**Figura 5.3** Esquema de coleta de dados em diagrama de blocos.

Ao observar a figura 5.3 pode-se ver que o bloco “step” é um gerador de função (para este caso, um gerador de degrau), onde o usuário pode selecionar o valor desejado. Em seguida, o bloco “Saturation” tem a finalidade de limitar a faixa de valores enviado para o equipamento, lembrando que o motor cc funciona de -10 v até 10 v. Os blocos “v”, “velocidade” e “To Workspace” servem para armazenar de forma gráfica e enviar os valores de tensão e velocidade para a área de trabalho do Matlab®respectivamente.

A comunicação entre o PC e o kit é feito pelos conversores D/A e A/D. No Simulink®

é preciso utilizar um driver de comunicação compatível com esse kit. Os blocos “Digital to Analog Converter” e “Analog to Digital Converter” são as representações virtuais desses conversores e são eles que fazem a comunicação entre o kit e o software. Estes blocos estão em “Bad Link”, pois o equipamento não está conectado ao computador. O sinal é feito em tensão, o próximo bloco (Gain), faz então a conversão de tensão para velocidade, dada em RPM. Deve-se observar que essa conversão é o ganho do sensor de velocidade que transforma cada 1000 RPM em 2,5 v. Por último, o bloco “Constante” envia um valor fixo para somar com a resposta de velocidade do motor cc. Esse valor é para compensar offset gerado pelo equipamento.

Antes de estipular um valor constante para corrigir o offset, é preciso fazer a identificação se há realmente erro. A figura 5.4, ilustra como foi feito o teste para observar se há offset no equipamento.



**Figura 5.4** Identificação do Offset no kit.

A figura 5.4 é um diagrama de blocos montado no Matlab®/Simulink. O sinal proveniente do kit chega ao computador em valores de tensão, estes são convertidos em RPM. Em seguida, o sinal vai para dois blocos, o primeiro é um display, onde mostra o valor do offset e o segundo (Scope) é visualizador gráfico.

Para a identificação do offset basta manter o motor cc sem funcionar, se houver algum valor diferente de 0 no display significa que está havendo um erro de sinal na comunicação. Ou seja, o software estaria recebendo algum valor de tensão proveniente do kit. Isso acontece nos equipamentos eletroeletrônicos normalmente por interferências magnéticas do meio externo e imprecisões de alguns componentes eletrônicos.

Ao fazer o teste o valor encontrado no display foi de 275 RPM, equivalente a 0,6875 volts. Em alguns processos menos de uma unidade de tensão poderia ser considerada insignificante, mas como os experimentos deste trabalho requerem precisão, é necessário considerar esse desvio.

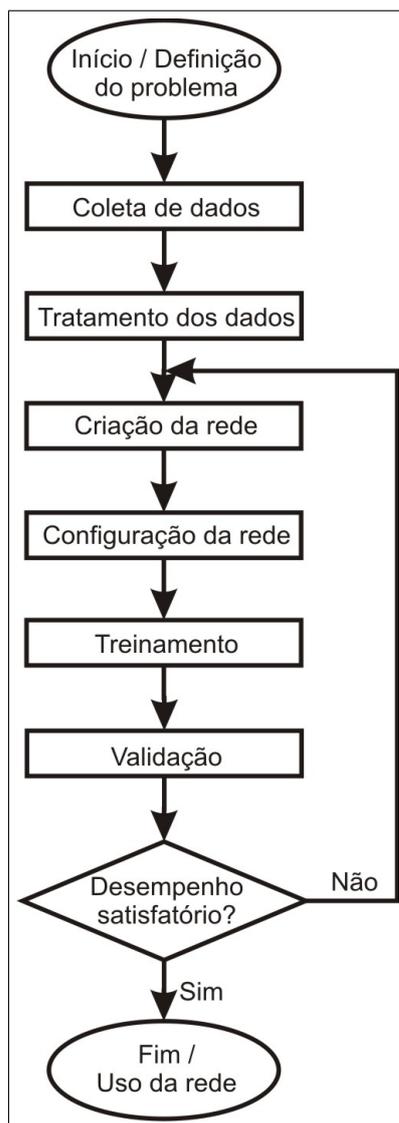
O passo que sucede a coleta de dados é o desenvolvimento da RNA. Para a descrição da metodologia de desenvolvimento foi reservado uma seção que será apresentada a seguir.

### 5.3 Desenvolvimento da RNA.

Segundo Beale et al. (2010), sete passos são sugeridos para o desenvolvimento de uma RNA. Estes passos foram adaptados no formato de fluxograma para melhor visualização do procedimento que será seguido. Veja a figura 5.5.

Na figura 5.5 percebe-se que após a definição do problema e coleta dos dados é estabelecida uma seqüência de passos para o projeto de uma RNA: tratamento dos dados, onde os mesmos são ajustados adequadamente dividindo em treinamento e teste. Esses dados são separados aleatoriamente, numa proporção pré definida, a fim de que a rede “aprenda” as regras e não “decore” exemplos. O próximo passo é a criação da rede, onde é escolhido qual rede utilizar para solucionar o problema, em seguida, sua configuração deverá ser feita. Depois deve-se fazer o treinamento da rede e, logo em seguida fazer a validação apresentando o restante dos dados à rede neural na fase de testes a fim de que ela possa “deduzir” corretamente o inter-relacionamento entre os dados. Com isso, se não forem obtidos resultados satisfatórios retorna-se etapa de criação da rede em busca de uma que melhor se aplique ao problema. Caso contrário, após a sua validação, a etapa final é o uso da rede.

A seguir será apresentada a necessidade do tratamento dos dados, bem como a forma como foi realizada.



**Figura 5.5** Passos para o desenvolvimento de uma RNA

### 5.3.1 Tratamento dos dados.

Segundo Beale et al. (2010) esta etapa tem a finalidade de manipular os dados colocando-os de forma mais adequada antes deles serem apresentados a rede. Inicialmente foi utilizado um algoritmo que, de acordo Carvalho et al. (2004), tem o propósito de reduzir a matriz de dados, inserir atrasos nas variáveis gerando novas entradas, normalização de acordo com as funções de ativação e, em seguida, fazer a divisão do conjunto inicial de amostras em dois novos conjuntos: o primeiro para o treinamento e o segundo para a validação. Esta divisão do conjunto de amostras foi feita na proporção de 70% e 30% para o conjunto de treinamento e validação res-

pectivamente. De acordo com Beale et al. (2010) uma das finalidades, por exemplo é transformar os dados no intervalo de  $[0; 1]$ , seguindo padrão da função de ativação. Outro exemplo é remover os vetores de entrada com os mesmo valores, diminuindo a matriz de dados. Ou ainda, recodifica dados desconhecidos (*NaN*).

A próxima subseção trata da criação da rede, essa etapa é onde se intensificam os estudos com o objetivo de definir qual rede usar para o problema de pesquisa em questão.

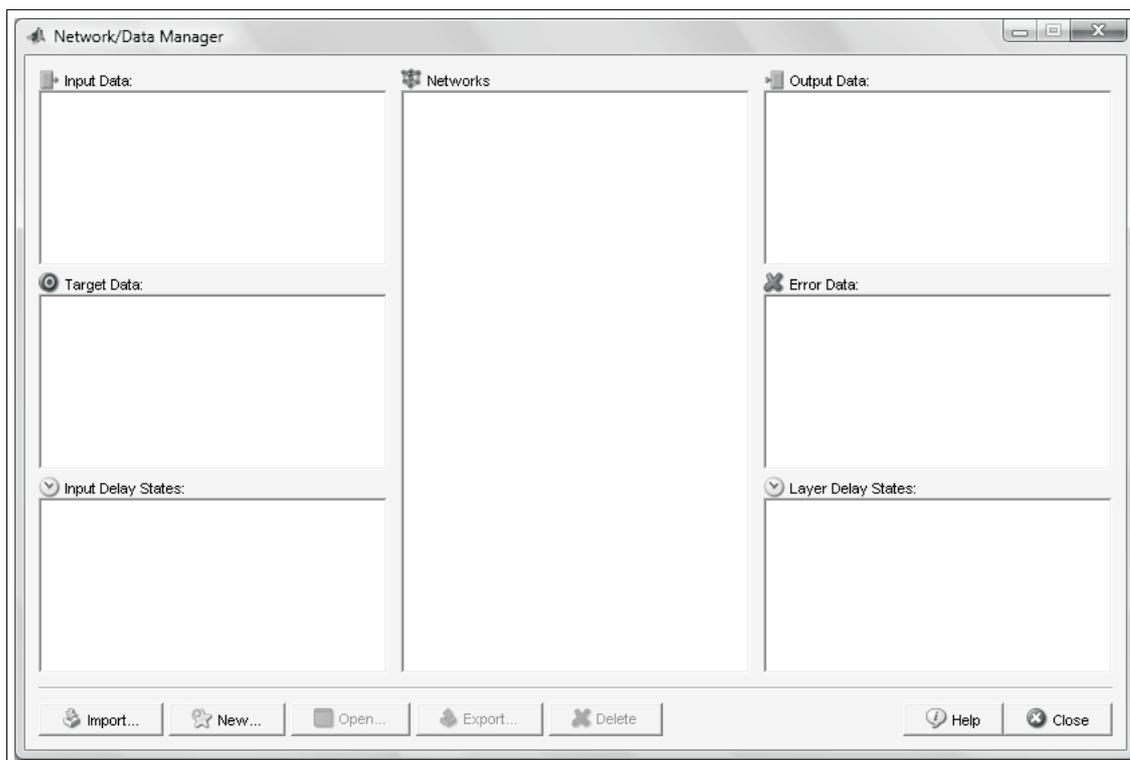
### 5.3.2 Criação da rede.

Reserva-se para esta etapa a atenção para a criação da rede, com isso, são feitos estudos de viabilidade de qual rede melhor está apta para a identificação do sistema real. Muitos são os tipos de redes neurais existentes para resolver problemas dos mais diversos. Diante dessa enorme possibilidade de escolha de rede, aparentemente fica difícil optar por um único modelo que solucione uma determinada aplicação desejada. De acordo Masters (1994), para esses casos, provavelmente poucos modelos fornecerão soluções excelentes.

Segundo Haykin (2001), para modelagem e identificação de sistemas dinâmicos o formato de redes neurais mais usadas são as de múltiplas camadas do tipo feed-forward (alimentadas à diante) utilizando o algoritmo de treinamento backpropagation. Pode-se dizer que o algoritmo backpropagation é uma generalização do método dos mínimos quadrados, que utiliza técnicas de gradiente descendente iterativo para minimizar uma função de custo igual ao erro médio quadrado entre a saída desejada e a saída da RNA. Outras topologias de rede poderiam ser usadas para o problema em questão. De acordo com Caner et al. (2004) uma das topologias que se aplicaria bem seria as redes de Função de Base Radial. Para este trabalho, optou-se por utilizar uma rede MLP com o algoritmo de treinamento backpropagation, conforme sugerido por Haykin (2001) e também por se deparar com diversas referências bibliográficas que apontavam ser um método relativamente simples de se projetar, eficiente e bastante difundido no meio acadêmico.

Toda etapa de criação e configuração, assim como o treinamento e validação da rede será feito no software MATLAB<sup>®</sup>, fazendo-se uso do toolbox de Redes Neurais

chamado nntool, o formato da janela que o usuário se depara ao fazer o uso desse toolbox pode ser visto na Figura 5.6.



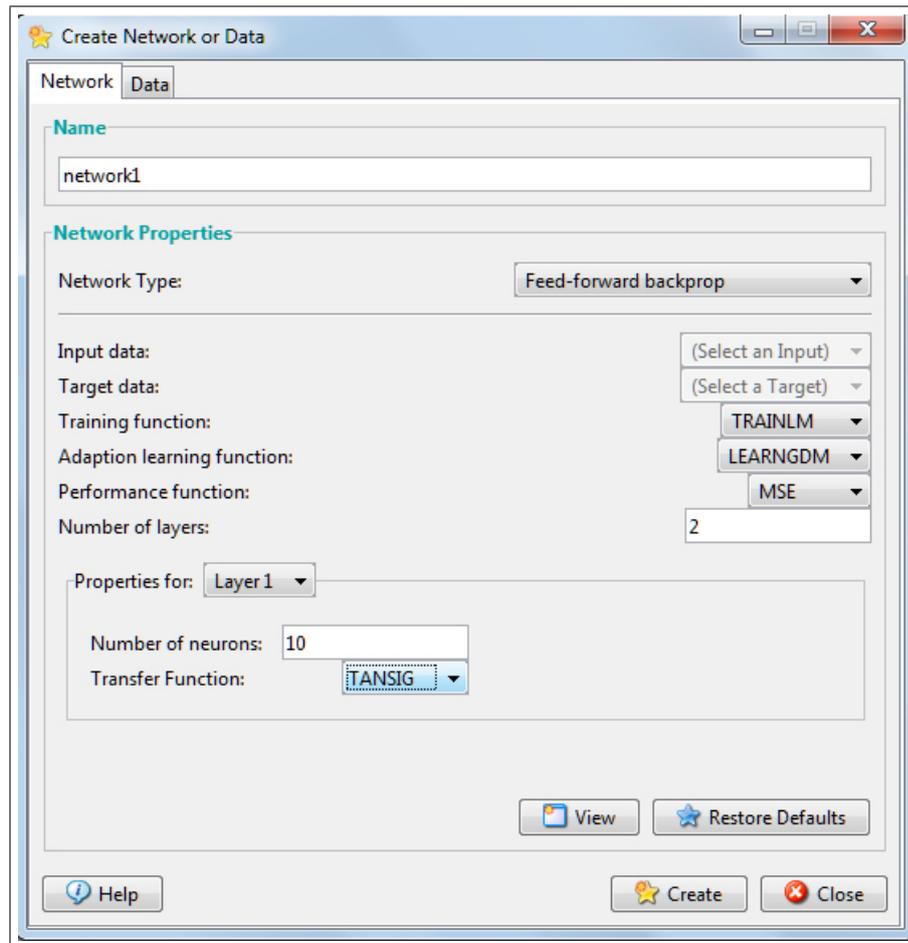
**Figura 5.6** Janela do nntool

A seguir serão apresentadas informações sobre a configuração da rede.

### 5.3.3 Configuração da rede.

A figura 5.7 representa a janela de configuração da RNA do toolbox. 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 observado para a validação da rede é o MSE (Mean Square Error) com a intenção de verificar a menor diferença entre o sistema e o modelo estimado. Ele foi escolhido por ter características aproximadas do algoritmo de treinamento usado. Sendo que o valor a ser usado como critério de parada é o mais próximo de zero desde que obedecido o número máximo de épocas para a convergência (epochs = 100).

Inicialmente a idéia é de que quanto mais camadas uma RNA possua, o seu desempenho tende a ser melhor, pois a sua capacidade de aprendizado seria mais a-



**Figura 5.7** janela de configuração RNA

primorada. Na prática, uma quantidade grande de camadas pode ser inviável, porque tornaria mais complexa a rede e aumentaria o seu esforço computacional, conseqüentemente maior seria o tempo demandado para resolução do problema. Conforme comentado previamente no capítulo 2, segundo Beale & Jackson (1992), basta usar apenas uma camada oculta e variar o número de neurônios nessa camada que já resolveria uma grande parte de problemas de generalização e de aproximação de funções. Na etapa de treinamento, será feita de forma empírica a investigação de qual rede se comporta melhor ao se variar o número de neurônios na camada oculta.

A subseção seguinte trata do treinamento da RNA. Esta etapa é onde ocorre a estimação dos valores de pesos e bias.

### 5.3.4 Treinamento da rede.

A função da etapa de treinamento é ajustar os valores das bias e dos pesos sinápticos até que a rede gere uma função generalizada capaz de solucionar o problema de modelagem do motor cc. A característica de aprendizado da rede utilizada será o supervisionado. Ou seja, será apresentada a rede valores de entrada e os valores desejados que o modelo deverá seguir. Só será considerada como um desempenho satisfatório quando houver uma diferença pequena entre o modelo estimado e a saída apresentada a rede.

O treinamento da rede foi feito para todos os valores de degrau inseridos no motor cc. Afim de que identificasse um modelo ideal para todas as faixas de valores de funcionamento do equipamento ao invés de algumas. Além disso, esse treinamento foi repetido várias vezes, variando-se o número de neurônio na camada oculta e catalogando o valor do MSE de cada excitação, gerando a tabela 5.2.

Com base nos dados da tabela 5.2 a proposta foi identificar o modelo com determinado número de neurônios que é o mais adequado para a modelagem. Inicialmente foi testado com 1 neurônio na camada oculta e em seguida foi aumentando este número até 10, o que gerou uma série de 100 treinamentos com seus respectivos valores de MSE. Então, foi extraída a média geométrica de cada série de excitações com seus respectivos números de neurônios na camada oculta. Segundo Costa e Silva et al. (2005) a média geométrica expressa melhor por apresentar, freqüentemente, distribuições irregulares devido a resultados discrepantes. Ao contrário da média aritmética que é influenciada pelos seus extremos. Os valores da media geométrica foram organizados na tabela 5.3.

Observando a tabela 5.3 pode-se verificar que a RNA que obteve um melhor de desempenho na modelagem foi a que continha em sua configuração apenas 3 neurônios na camada oculta. É importante deixar claro que os modelos RNA que possuíam 7, 8, 9 e 10 neurônios na camada oculta também obtiveram valores praticamente iguais ao modelo de 3 neurônios, isso se considerarmos o tão pequeno está a média geométrica do MSE desses modelos estimados. Então, não haveria problema se as utilizasse como rede ideal, mas como a RNA com 3 neurônios na camada oculta é a mais simples de se implementar optou-se por utilizá-la.

**Tabela 5.2** MSE para os diferentes números de neurônios na camada oculta e diferentes estímulos de entrada.

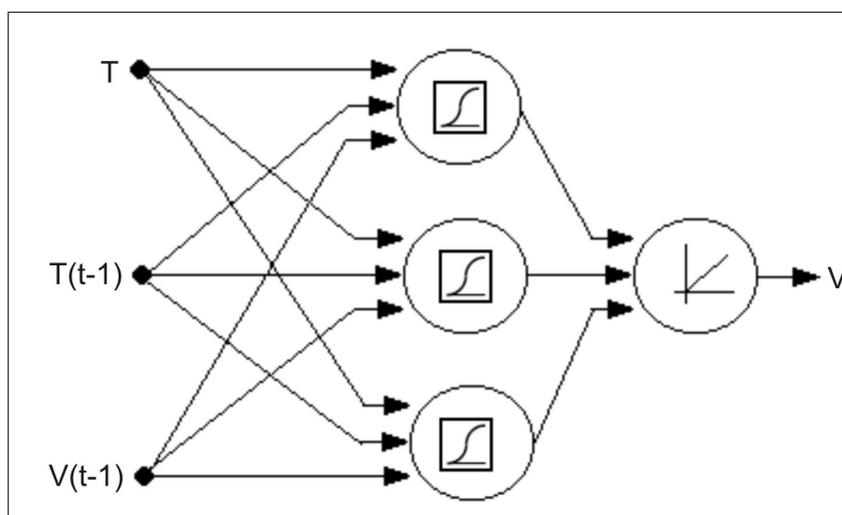
D \ N	1	2	3	4	5	6	7	8	9	10
-2	1,29E-05	3,27E-06	1,49E-06	4,12E-05	1,22E-05	1,31E-05	3,79E-05	2,57E-06	1,75E-05	1,01E-05
-4	1,10E-04	7,73E-05	1,27E-04	8,23E-04	1,05E-04	7,32E-04	0,0027	8,49E-05	7,01E-05	7,70E-05
-6	3,21E-04	2,29E-06	0,0050	1,04E-06	0,0028	3,47E-04	1,77E-04	1,37E-06	1,05E-05	2,25E-06
-8	8,90E-05	0,0016	0,0049	1,79E-04	0,0086	6,85E-05	3,32E-05	0,0021	3,71E-05	4,56E-05
-10	0,0045	1,31E-05	0,0013	7,77E-06	9,77E-06	1,77E-05	1,14E-04	8,32E-06	1,00E-05	1,25E-05
2	5,56E-05	3,98E-04	1,32E-08	5,57E-08	7,06E-05	1,76E-05	4,47E-04	1,37E-05	2,78E-06	2,42E-11
4	4,17E-06	2,07E-05	2,12E-06	2,41E-06	1,74E-05	1,80E-05	4,10E-05	2,24E-05	0,0049	4,02E-05
6	6,78E-09	3,29E-07	1,41E-07	3,25E-07	1,94E-04	3,76E-04	3,13E-07	2,91E-07	1,20E-06	3,84E-05
8	2,34E-05	2,34E-05	1,17E-05	7,23E-05	2,04E-06	1,31E-06	1,28E-05	1,42E-05	1,79E-05	4,92E-04
10	2,22E-06	2,42E-06	6,96E-07	1,56E-07	2,04E-06	3,43E-05	7,37E-06	4,71E-07	4,33E-07	1,55E-04

D - Degrau de tensão inserido no motor cc N - Neurônios na camada oculta da rede

**Tabela 5.3** Média geométrica do MSE para cada valor de neurônios na rede.

Número de neurônios	Média geométrica do MSE
1	1,14E-05
2	1,02E-05
3	1,80E-06
4	1,09E-05
5	4,06E-05
6	2,71E-05
7	8,02E-06
8	4,85E-06
9	7,93E-06
10	8,81E-06

Um exemplo do modelo de RNA utilizada é apresentada na figura 5.8. Observe-se que a camada oculta da rede é composta de três neurônios usando a função de ativação do tipo logarítmica sigmóide (logsig) e a camada de saída contém apenas um neurônio com a função do tipo linear (pureline). As letras T e V representam a tensão e a velocidade apresentadas à rede respectivamente.

**Figura 5.8** RNA do tipo MLP (adaptado de Carvalho et al. (2004))

Encontrados valores para os pesos e bias que possibilitam a rede generalizar o modelo de forma mais próxima do real com o conjunto de treinamento. A seção seguinte fala sobre a validação da rede no momento em que a RNA é submetida ao teste com o conjunto de validação.

### 5.3.5 Validação da rede.

Nesta etapa, os 30% do conjunto de dados restantes que foram destinados para esse fim são “atravessados” a rede com o intuito de validar a RNA escolhida na etapa de treinamento. Com isso, nenhum parâmetro foi modificado na rede usada no treinamento, ou seja, foi mantida toda a configuração da rede.

Na etapa de validação se a RNA também obtiver pequenos valores do MSE a rede vai ser considerada própria para o uso. Caso contrário, será necessário fazer novamente outras configurações ou, se for necessário, criar outro tipo diferente de RNA.

Para um sistema digital é necessário que o modelo do motor cc seja apresentado no formato discreto, pois até o momento o mesmo estava no domínio da frequência. A seguir a obtenção deste modelo discreto será comentada.

## 5.4 Obtenção do modelo discreto do motor cc.

Até o momento todo o procedimento de modelagem matemática do motor cc foi feita com base na Teoria de Controle Clássico. A função de transferência obtida é apresentada no domínio da frequência, trabalhando em tempo contínuo. Como os experimentos são feitos de forma digital, utilizando um microcomputador para estimação dos parâmetros, fica clara a necessidade de aplicação das técnicas de Controle Digital. Segundo Ogata (2004) sistemas de tempo discreto diferem dos de tempo contínuo, pois os sinais para um sistema de tempo discreto estão na forma amostrada.

A inclusão de um computador digital em um sistema analógico produz sinais em forma digital, em parte do sistema. O sistema toma a forma de uma combinação mista digital-analógica. A introdução de um computador digital em um sistema de controle requer o uso de conversores digital-analógico e analógico-digital. A conversão de um sinal analógico para o correspondente sinal digital é uma aproximação porque o sinal analógico pode assumir um número infinito de valores, ao passo que a variedade de diferentes números que podem ser formados por um conjunto finito de dígitos é limitada.

Para encontrarmos o modelo discreto de uma função de transferência é preciso utilizar o método da transformada z. De acordo com Ogata (2004) o papel desempenhado pela transformada z em sistema de tempo discreto é bastante similar ao da transformada de Laplace em sistemas de tempo contínuo. Para a obtenção da função de transferência em tempo discreto foi usado o mapeamento hold equivalence. Para maiores informações sobre essa metodologia de discretização consultar Franklin et al. (1990). O modelo discreto da equação (4.2) é apresentado em (5.1).

$$\frac{\Omega(z)}{V(z)} = \frac{K(1 - e^{-\frac{T}{\tau}})}{Z - e^{-\frac{T}{\tau}}} \quad (5.1)$$

A equação de recorrência que corresponde a (5.1) é a equação (5.2).

$$\omega(kT) = a \cdot \omega(kT - T) + b \cdot u(kT - T) \quad (5.2)$$

Onde  $a$  e  $b$  são os parâmetros que precisam ser estimados e representam respectivamente a  $K(1 - e^{-\frac{T}{\tau}})$  e ainda a  $e^{-\frac{T}{\tau}}$ .

Encontrado o modelo discreto do motor cc é possível fazer a estimação dos parâmetros dele pelos Mínimos Quadrados. A próxima seção fala sobre o método em questão.

## 5.5 Estimação pelo Método dos Mínimos Quadrados.

Uma maneira de processar as medidas seria pelo método conhecido como mínimos quadrados em lote (KUGA, 2005). Esse nome é dado porque as medidas são processadas no mesmo instante.

Considerando que temos dois parâmetros a serem estimados e lembrando que  $m$  é o número de medidas, pode-se observar que:

- Se  $m = 1$ , solução indeterminada.

- Se  $m = 2$ , então:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} t_1 & 1 \\ t_2 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} \quad (5.3)$$

Resultando,

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} t_1 & 1 \\ t_2 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad (5.4)$$

Se  $m > 2$ , então a solução é encontrada utilizando a equação normal. Para isso, basta multiplicar ambos os lados da equação ( $y = Hx$ ) por  $H^t$ , resultando:

$$H^t \vec{y} = H^t \cdot H \vec{x} \quad (5.5)$$

O capítulo a seguir apresenta os resultados encontrados através dos dois métodos sugeridos na dissertação.

## Capítulo 6

### ANÁLISE DOS RESULTADOS

#### 6.1 Topologia da RNA escolhida.

Ao efetuar os experimentos para validação da rede, concluiu-se que a RNA do capítulo anterior era a mais adequada, por apresentar menor média de erro em todas as faixas de valores de tensão para o funcionamento do motor. Então, seus dados foram armazenados e serão apresentados mais adiante de forma gráfica e numérica, através de uma tabela com os seus respectivos MSE (Mean Square Error).

Segundo Navarro et al. (2010) para cada tipo de RNA existe seu modelo matemático correspondente. O que representa a rede utilizada nesse trabalho, com as características citadas anteriormente é o que segue,

$$\begin{cases} N_1 = \frac{1}{e^{-(\sum x_i \cdot w_{ij} + b_j)}} \\ N_2 = \frac{1}{e^{-(\sum x_i \cdot w_{ij} + b_j)}} \\ N_3 = \frac{1}{e^{-(\sum x_i \cdot w_{ij} + b_j)}} \end{cases} \quad (6.1)$$

referente a camada oculta, e

$$N_4 = y = \sum N_j \cdot w_{ij} + b_j \quad (6.2)$$

referente a camada de saída.

Onde  $x_i$  é a  $i$ -ésima entrada, que comparada a figura 5.8 corresponde a  $T$ ,  $T(t-1)$  e  $V(t-1)$ .  $y$  a saída, que corresponde a  $V$ .  $w_{ij}$  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.  $b_j$

**Tabela 6.1** Valores encontrados de Pesos e bias para a RNA (Fonte: Navarro et al. (2010)).

	$X_1$	$X_2$	$X_3$	Bias
$N_1$	$w_{11} = 5.7201$	$w_{21} = 5.5852$	$w_{31} = 7.7122$	$b_1 = -9.8313$
$N_2$	$w_{12} = 6.2787$	$w_{22} = -3.5577$	$w_{32} = 3.1776$	$b_2 = -3.1282$
$N_3$	$w_{13} = -11.9332$	$w_{23} = 27.0293$	$w_{33} = -7.8961$	$b_3 = -8.253$
$N_4$	$w_{14} = 0.63992$	$w_{24} = 0.29615$	$w_{34} = 0.31384$	$b_4 = -0.012543$

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  $j = 1, 2, 3, 4$ .

No instante inicial do treinamento o próprio software arbitra valores para os pesos e bias para que eles sejam modificados quando treinados. Em seguida, ele faz uma busca até encontrar valores para estes parâmetros que proporcionam pequenos erros no processo de modelagem. Os valores de pesos e bias encontrados são os apresentados na tabela 6.1.

Após definida exatamente qual RNA utilizar através dos testes no capítulo anterior. A próxima seção mostra os resultados da estimação por RNA.

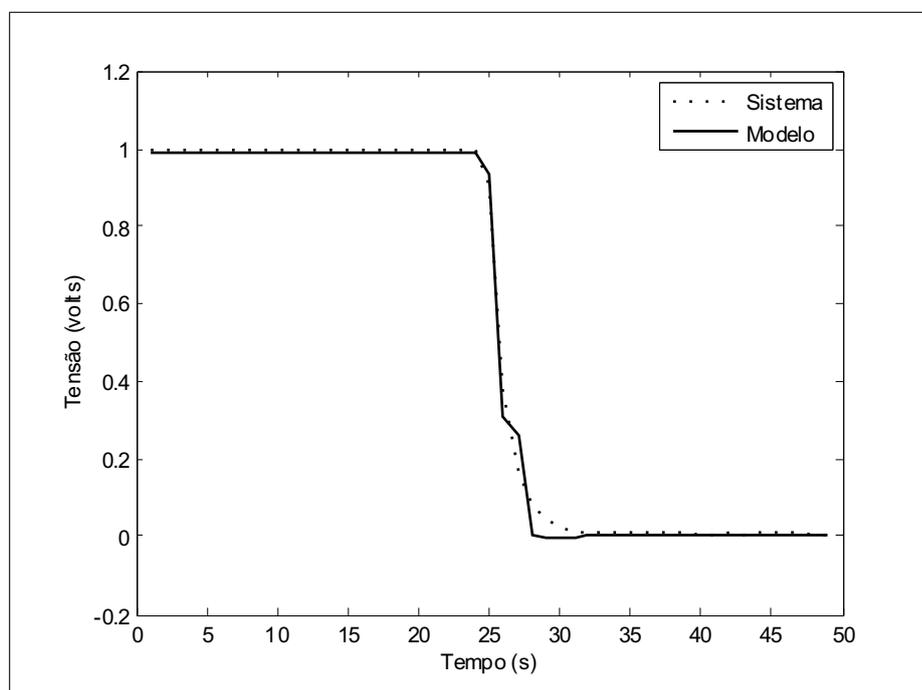
## 6.2 Resultado da modelagem via RNA.

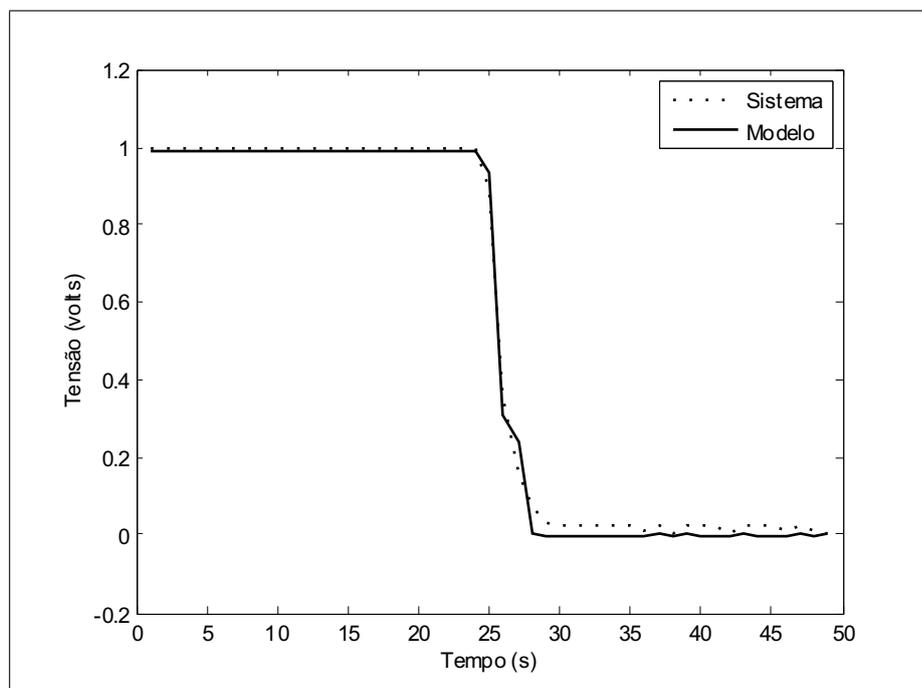
O desempenho dos modelos estimados por RNA foram compilados e serão apresentados numa tabela. Em seguida, os gráficos que comparam a resposta do sistema em questão para os diversos degraus de tensão aplicados e o modelo estimado pela rede são apresentados. Seguem na tabela 6.2 os valores do MSE encontrado na etapa de validação.

Percebe-se novamente que também na etapa de validação os valores dos MSE para cada conjunto de amostra ficaram muito baixo. Com esse resultado podemos concluir que o modelo realmente representa com precisão um motor cc no formato de RNA. As figuras 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9 e 6.10 são os gráficos que comparam o modelo estimado com o sistema real para cada degrau de tensão aplicado ao motor cc.

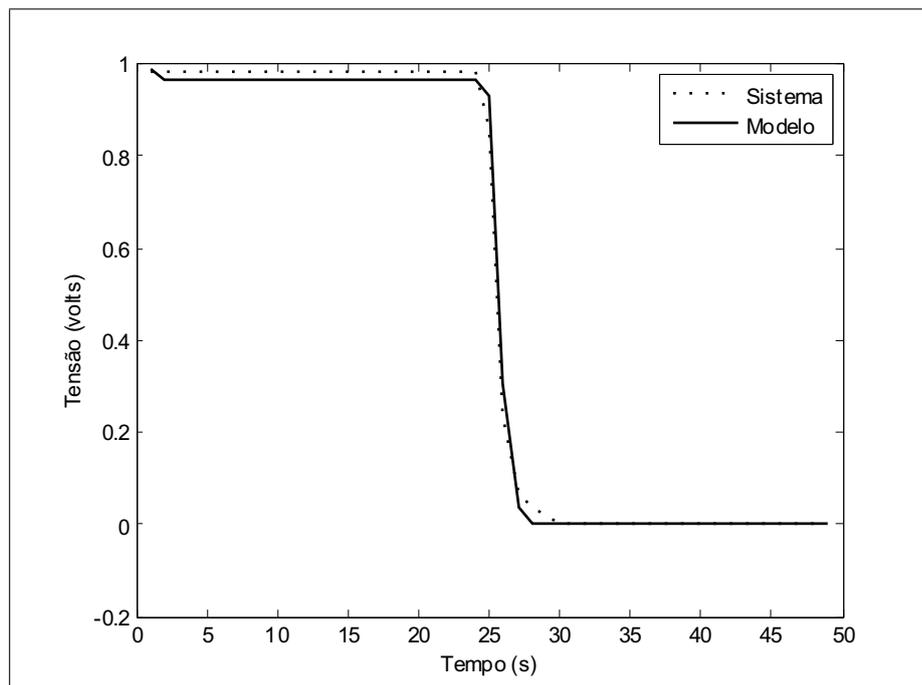
**Tabela 6.2** Valores do MSE para as diversas amplitudes de degrau.

Degrau	Conjunto de dados	MSE
-10V	Validação	6,3447e-004
-8V	Validação	6,3549e-004
-6V	Validação	4,1558e-004
-4V	Validação	3,9008e-004
-2V	Validação	5,1851e-004
2V	Validação	0,0011
4V	Validação	0,0016
6V	Validação	0,0020
8V	Validação	0,0038
10V	Validação	0,0038

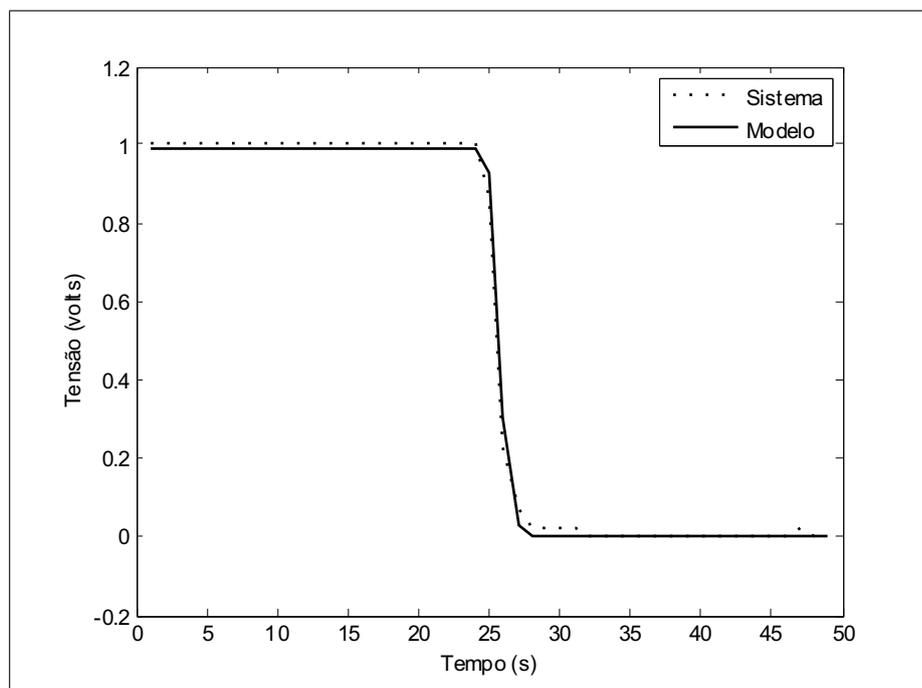
**Figura 6.1** Modelo RNA versus Sistema real (degrau -10V).



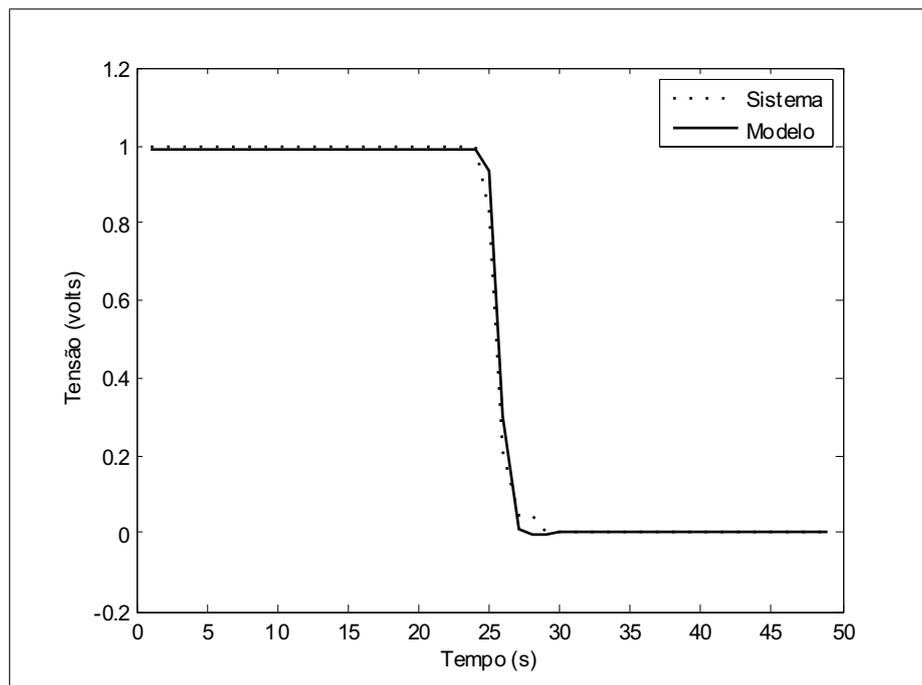
**Figura 6.2** Modelo RNA versus Sistema real (degrau -8V).



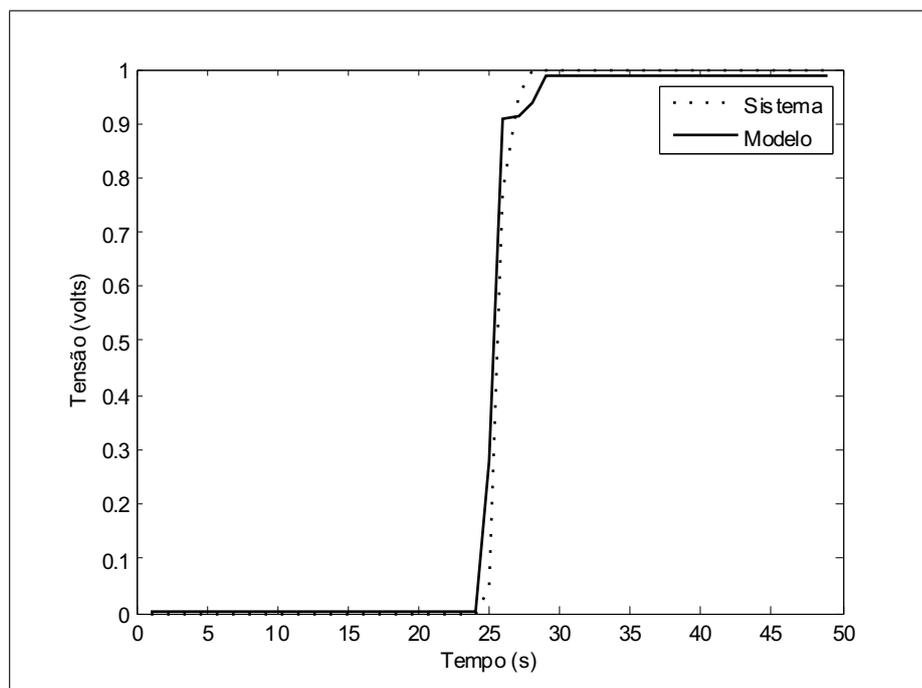
**Figura 6.3** Modelo RNA versus Sistema real (degrau -6V).



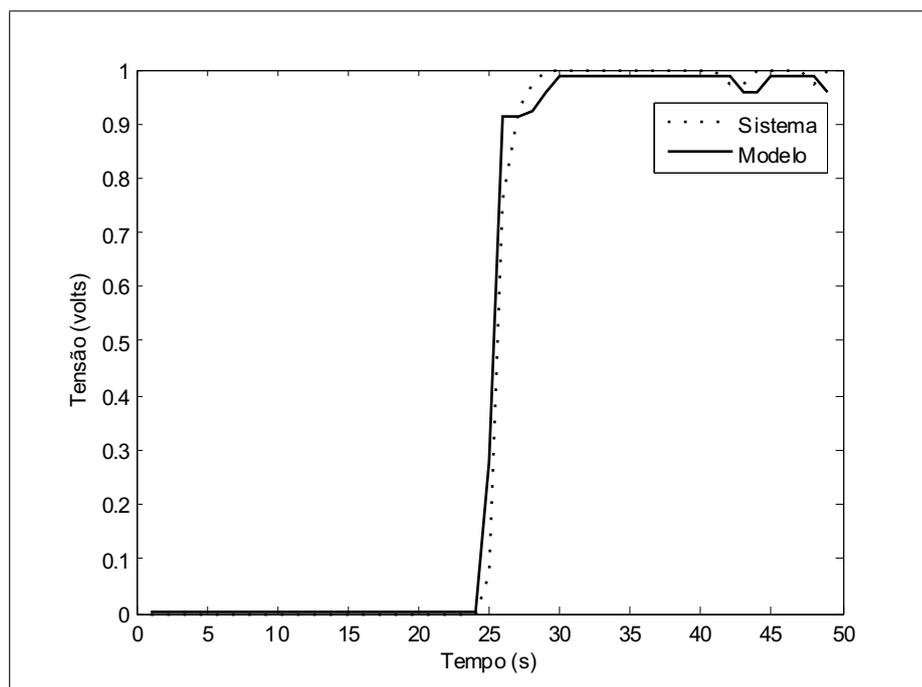
**Figura 6.4** Modelo RNA versus Sistema real (degrau -4V).



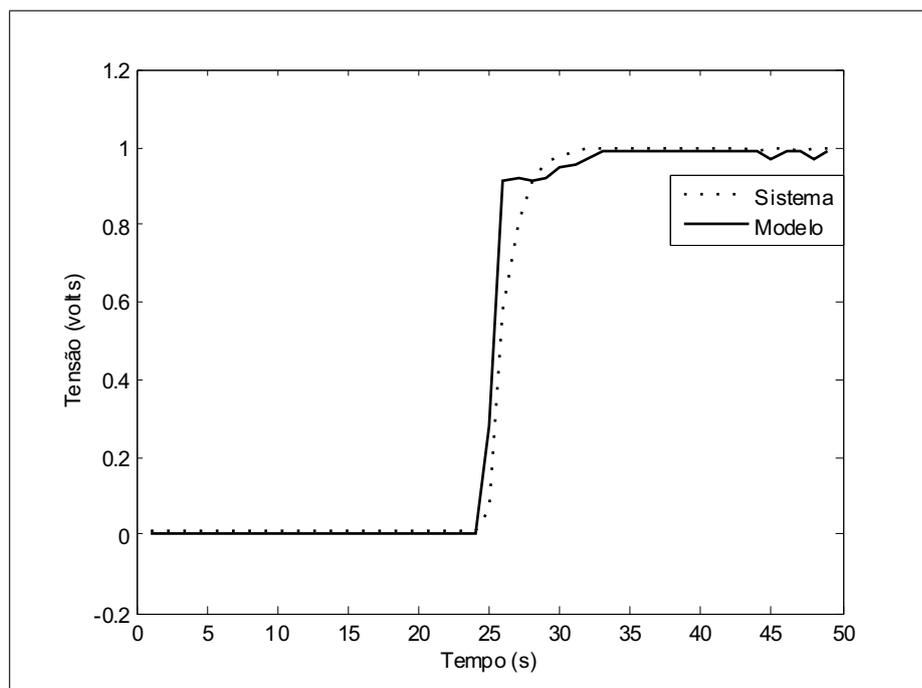
**Figura 6.5** Modelo RNA versus Sistema real (degrau -2V).



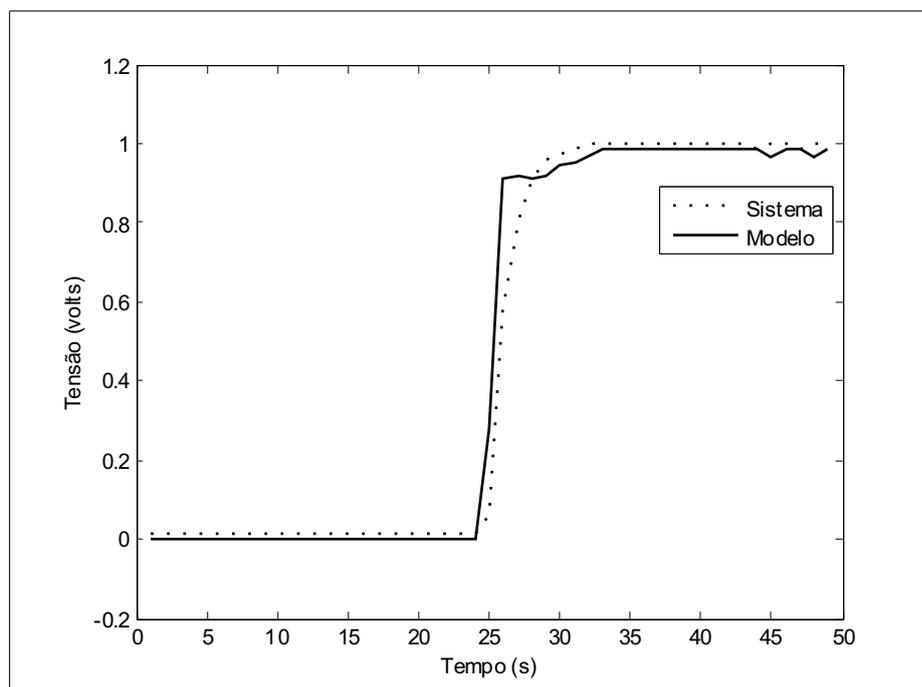
**Figura 6.6** Modelo RNA versus Sistema real (degrau 2V).



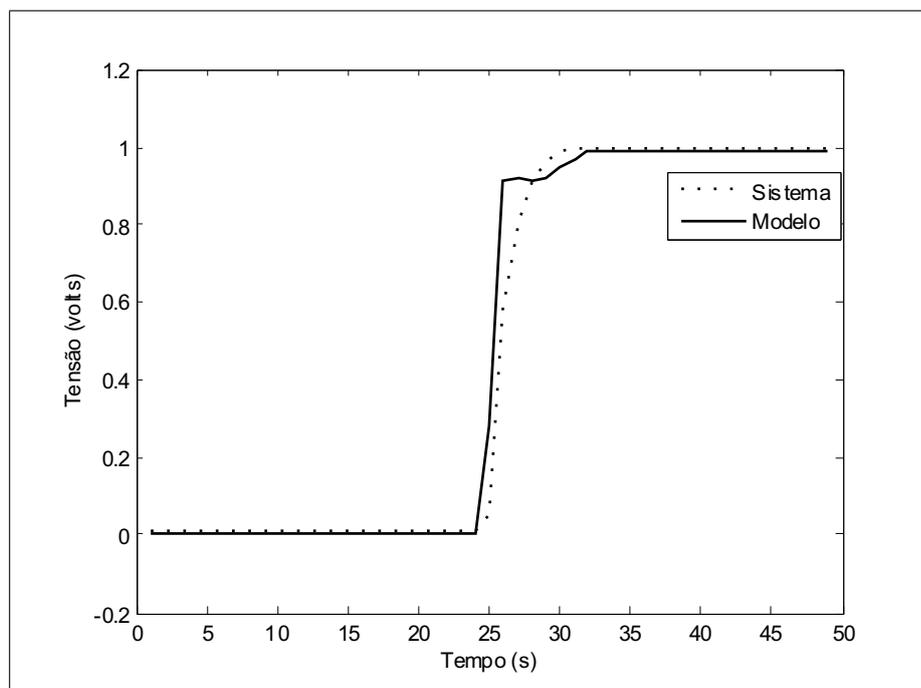
**Figura 6.7** Modelo RNA versus Sistema real (degrau 4V).



**Figura 6.8** Modelo RNA versus Sistema real (degrau 6V).



**Figura 6.9** Modelo RNA versus Sistema real (degrau 8V).



**Figura 6.10** Modelo RNA versus Sistema real (degrau 10V).

A seguir os resultados pelos Mínimos Quadrados serão apresentados.

### 6.3 Resultado da estimação pelo Método dos Mínimos Quadrados.

Como a quantidade de medidas era muito grande, ou seja, o valor de  $m$  era muito grande, ficou claro que para a estimação pelo método dos Mínimos Quadrados, os cálculos precisam ser feitos em lote. Para isso, um algoritmo no Matlab® foi elaborado.

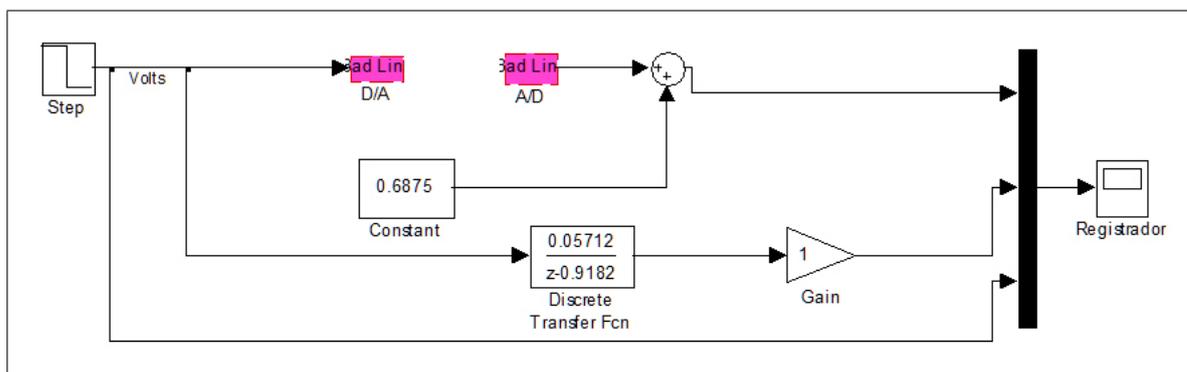
O resultado da estimação dos parâmetros  $a$  e  $b$  para cada degrau de tensão foi compilado e apresentado na tabela 6.3.

Tendo em mãos os valores dos parâmetros  $a$  e  $b$ , conforme tabela 6.3, é preciso conhecer um valor médio deles que corresponda com a maior fidelidade possível o motor cc, em todas as suas faixas de operação. Para isso, foi extraída uma média aritmética simples do resultado da estimação. Os valores encontrados para cada parâmetro foram  $a = 0,057117$  e  $b = 0,91825$ . Esses valores foram substituídos no modelo matemático do motor cc, assim é possível fazer o experimento para encontrar o MSE de cada degrau de tensão.

**Tabela 6.3** Parâmetros encontrados para cada degrau.

Degrau	Parâmetro (a)	Parâmetro (b)
-10V	0,03035	0,9569
-8V	0,08425	0,9020
-6V	0,08467	0,7419
-4V	0,05924	0,9401
-2V	0,06535	0,9276
2V	0,05279	0,9487
4V	0,04928	0,9454
6V	0,09801	0,8878
8V	0,02618	0,9661
10V	0,02105	0,9660

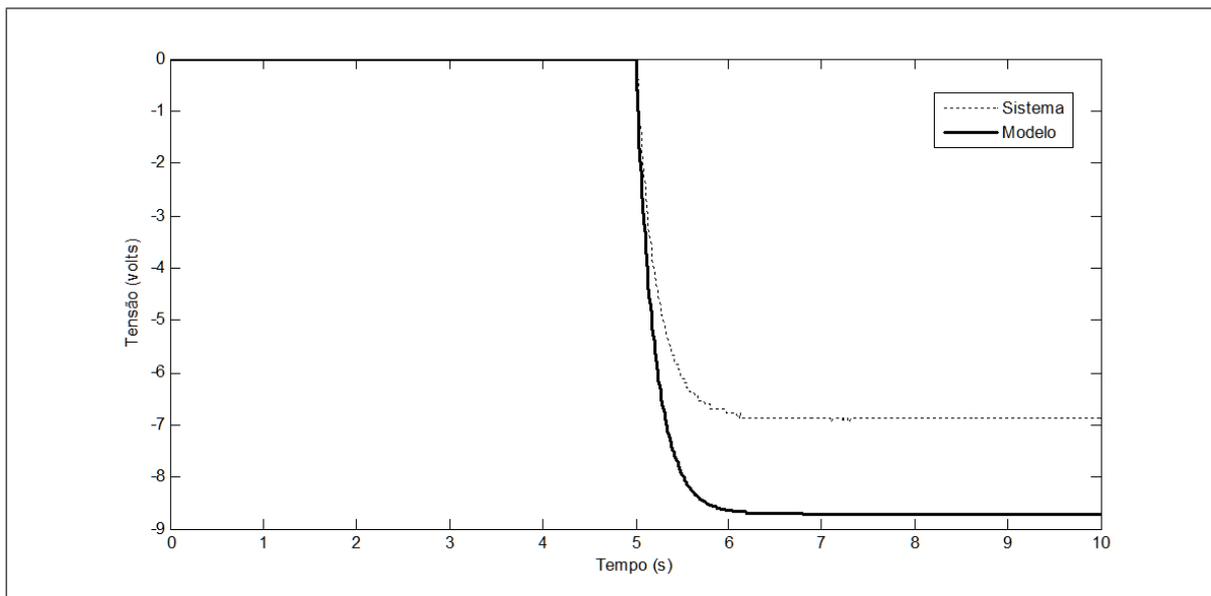
O experimento foi feito de acordo com a figura 6.11. A mesma representa a tela do Matlab®/Simulink, que possibilita fazer simulações com equipamentos externos em tempo real através de diagramas de blocos. Ao analisar a figura vemos que o valor de tensão é enviado para o conversor D/A que comunica com o motor cc, sua resposta é recebida pelo computador através do conversor A/D. Essa resposta é armazenada no bloco registrador. Esse mesmo degrau é também enviado para a função de transferência do motor cc e sua resposta é armazenada no registrador para que posteriormente seja feito o cálculo do MSE. O resultado desse cálculo se encontra na tabela 6.4.

**Figura 6.11** Experimento pelo Simulink

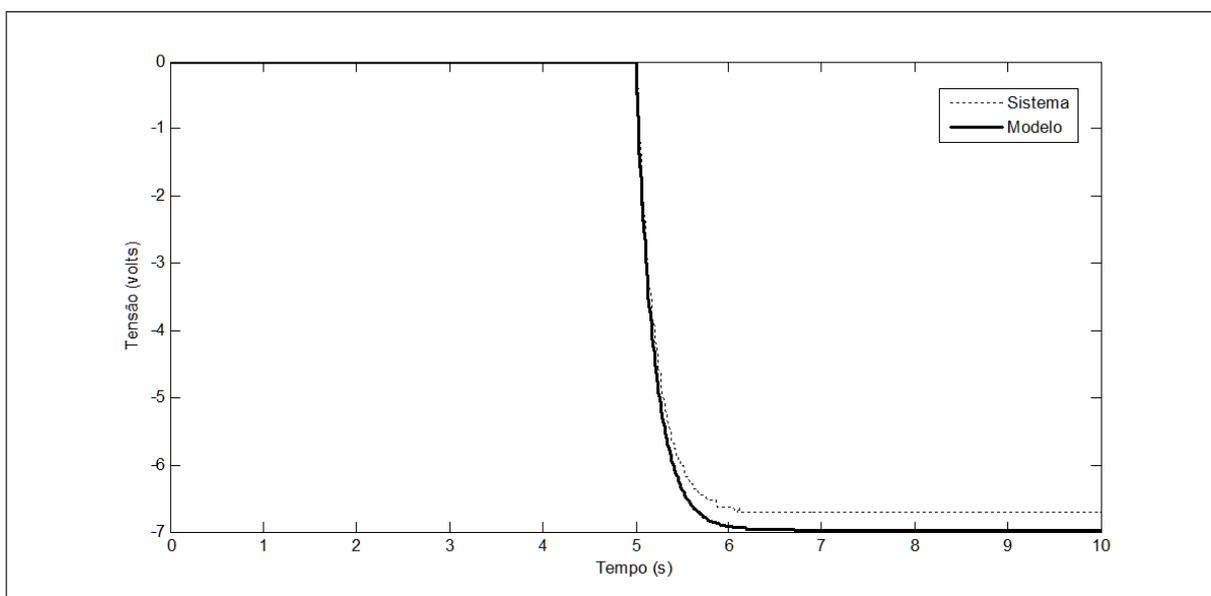
**Tabela 6.4** MSE encontrados pelos Mínimos Quadrados.

<b>Degrau</b>	<b>MSE</b>
-10V	1,6195
-8V	0,0370
-6V	0,0105
-4V	0,0146
-2V	0,0149
2V	0,0049
4V	0,0082
6V	0,0128
8V	0,3596
10V	3,1751

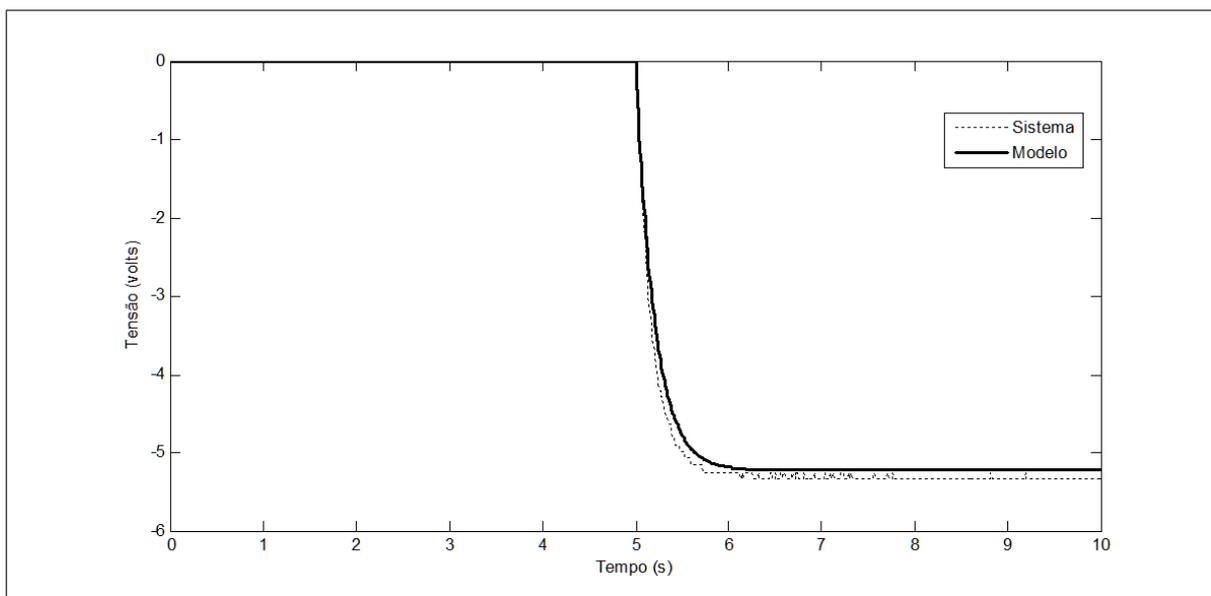
Ao analisar a tabela 6.4 fica claro que o MSE mesmo com valores maiores que os encontrados pela estimação via RNA, possui resultados bastante pequenos. O teste de qualidade de ajuste ( $r^2$ ) foi feito e encontrado  $r^2 = 0,99$ , o que conclui que o modelo se ajusta muito bem aos dados. O experimento feito na figura 6.11 produziu os gráficos que comparam a resposta do motor cc para os diversos degraus de tensão aplicados e o modelo estimado pelos Mínimos Quadrados. As figuras 6.12, 6.13, 6.14, 6.15, 6.16, 6.17, 6.18, 6.19, 6.20 e 6.21 são os resultados do experimento feito por esse método de estimação.



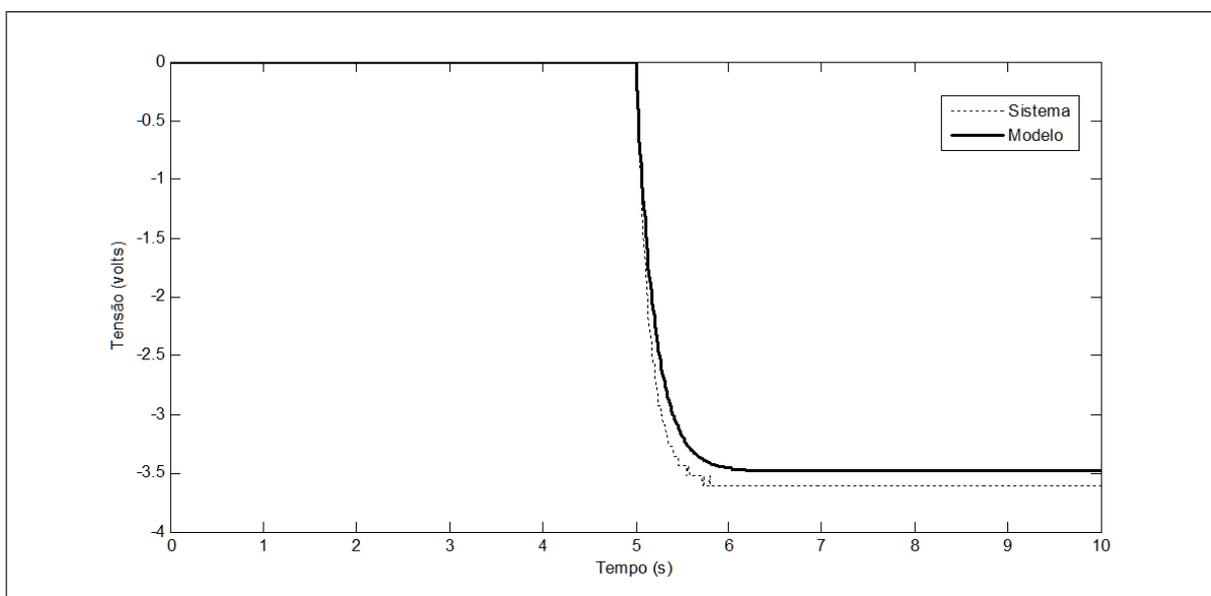
**Figura 6.12** Modelo versus Sistema real (degrau -10V).



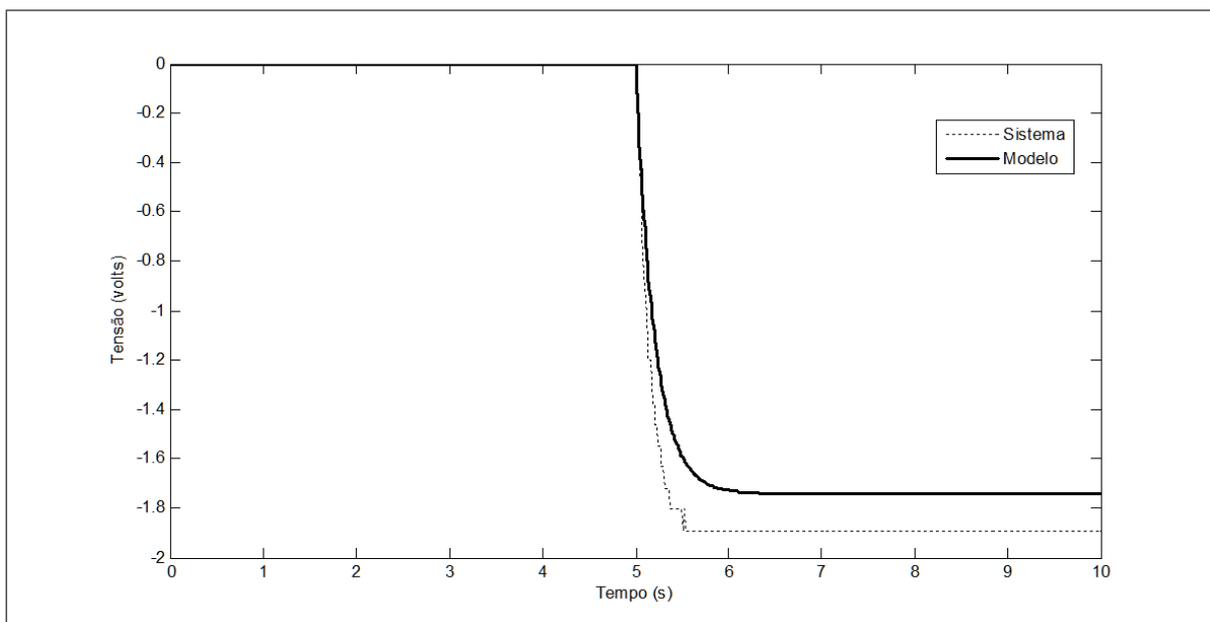
**Figura 6.13** Modelo versus Sistema real (degrau -8V).



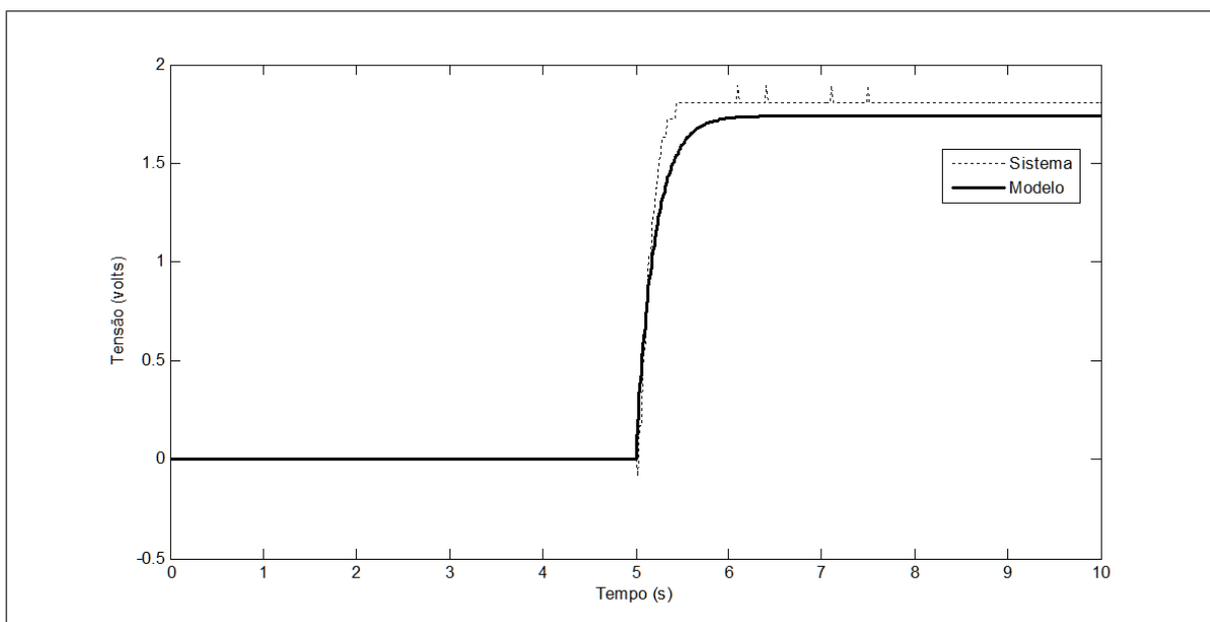
**Figura 6.14** Modelo versus Sistema real (degrau -6V).



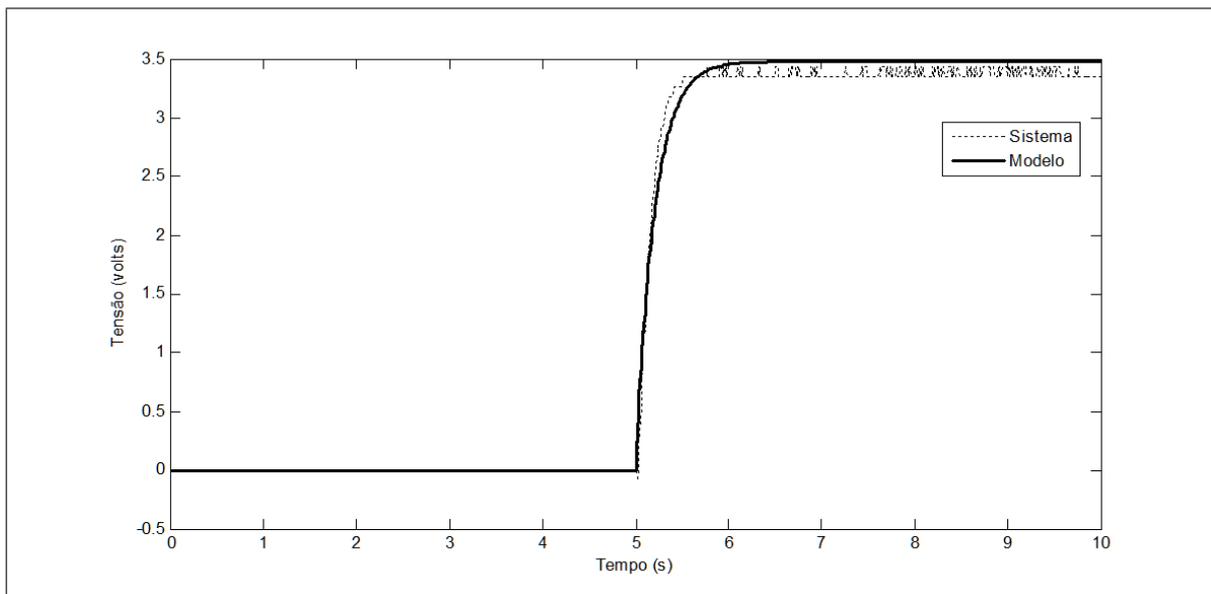
**Figura 6.15** Modelo versus Sistema real (degrau -4V).



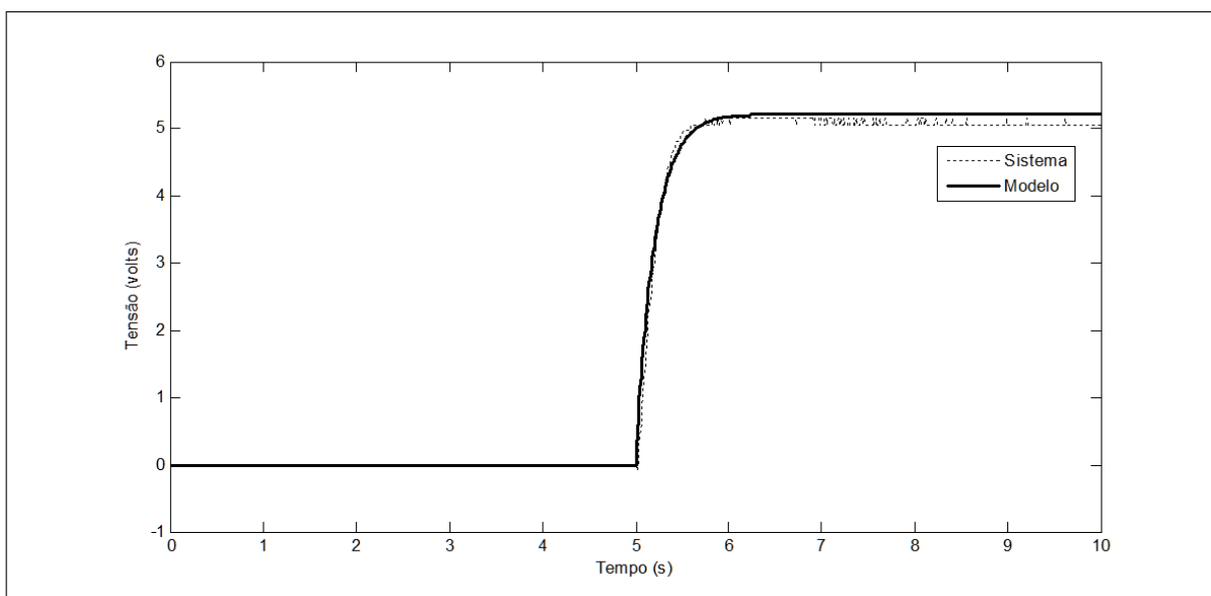
**Figura 6.16** Modelo versus Sistema real (degrau -2V).



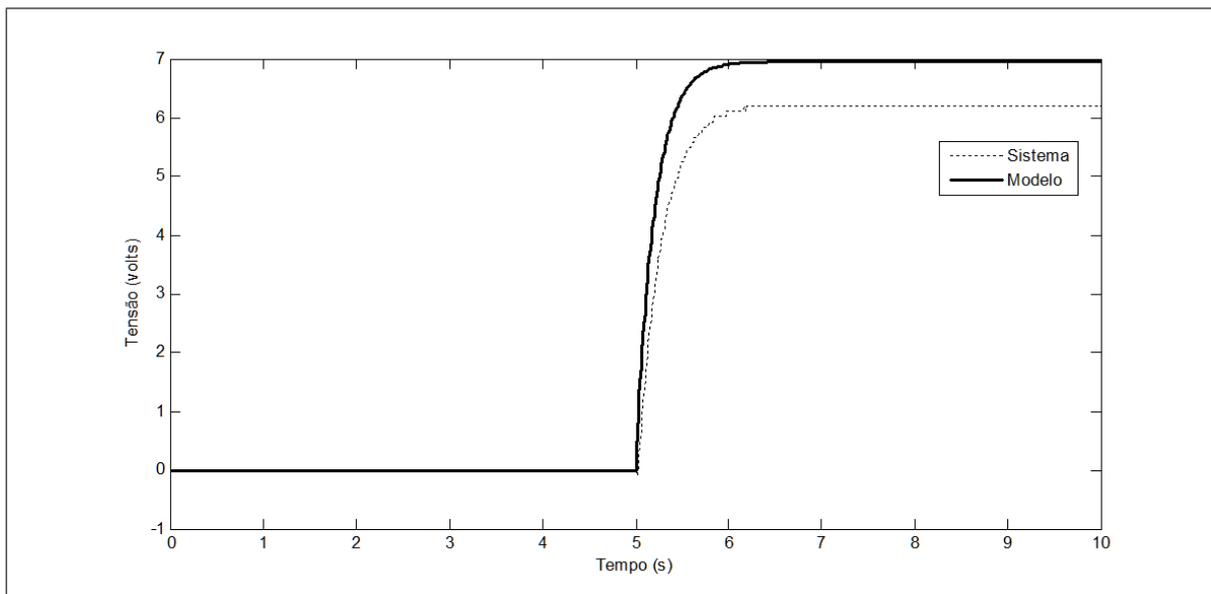
**Figura 6.17** Modelo versus Sistema real (degrau 2V).



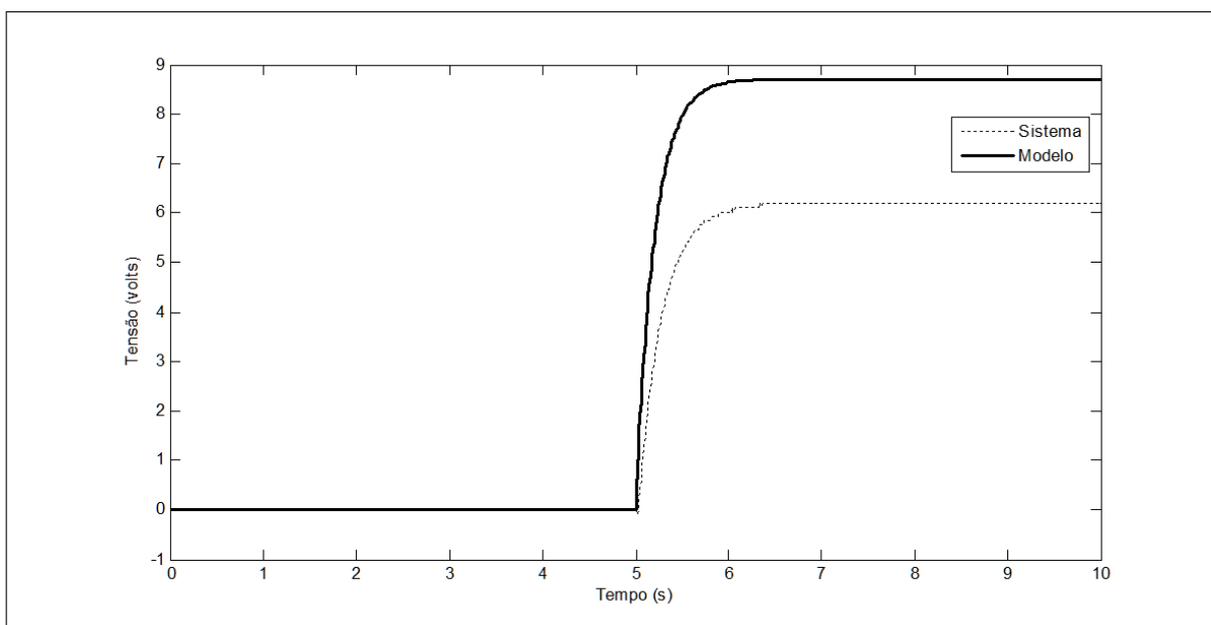
**Figura 6.18** Modelo versus Sistema real (degrau 4V).



**Figura 6.19** Modelo versus Sistema real (degrau 6V).



**Figura 6.20** Modelo versus Sistema real (degrau 8V).



**Figura 6.21** Modelo versus Sistema real (degrau 10V).

O próximo capítulo é destinado a conclusão de todo o processo de estimação pelos métodos sugeridos, bem como a comparação entre esses métodos. O capítulo ainda apresenta sugestões a respeito de possíveis trabalhos que possam vir a acrescentar a dissertação proposta.

## Capítulo 7

# CONSIDERAÇÕES FINAIS

### 7.1 Conclusão.

Nesta dissertação foi apresentado um estudo comparativo de dois métodos de identificação de modelos dinâmicos. O sistema modelado no trabalho foi um motor cc do kit didático chamado Servo Fundamentals Trainer. Com esse estudo obteve-se resultados muito próximo do sistema real, tanto usando RNA quanto utilizando o método de Mínimos Quadrados. Essa observação comprova que a escolha de um ou outro atenderá a exigência de maior fidelidade possível ao processo.

Em primeiro momento, foi investigado o referencial teórico a respeito das duas metodologias para se obter mais detalhes de embasamento dessas bases científicas. Em seguida, principalmente para explicitar os fatos que motivaram a presente pesquisa e sua adequação, é apresentado o estado da arte da identificação de sistemas dinâmicos usando RNA bem como o método de Mínimos Quadrados. Por último, é descrita a metodologia e os resultados alcançados por este caminho.

As análises de desempenho foram extraídas comparando-se o MSE e ainda as discrepâncias das curvas do modelo estimado em relação ao modelo físico. Os resultados estão dispostos na tabela 7.1.

Com base nos resultados apresentados na tabela 7.1 é possível notar que a RNA foi mais eficiente na identificação do que o método de Mínimos Quadrados em todos os degraus de tensão aplicados ao motor cc. É importante observar que os Mínimos Quadrados também encontraram valores bastante satisfatórios principalmente quando aplicado pelo menos 50% do degrau de tensão em seus terminais. Ou seja, para uma requisição instantânea de velocidade até a metade de sua capacidade máxima os valores do MSE ficaram com no máximo um centésimo. Para equipamentos como este é um resultado bastante razoável. Estes resultados mostraram a aplicabilidade do método dos Mínimos Quadrados quando se deseja ajustar um conjunto de dados

**Tabela 7.1** Comparação dos valores do MSE pelos dois métodos propostos.

Degrau	MSE - Mínimos Quadrados	MSE - RNA
-10V	1,6195	6,3447e-004
-8V	0,0370	6,3549e-004
-6V	0,0105	4,1558e-004
-4V	0,0146	3,9008e-004
-2V	0,0149	5,1851e-004
2V	0,0049	0,0011
4V	0,0082	0,0016
6V	0,0128	0,0020
8V	0,3596	0,0038
10V	3,1751	0,0038

experimentais.

Na identificação por RNA, para o mesmo conjunto de dados os valores do MSE ficaram ainda menores. 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. Além disso, as redes neurais também são capazes de prever o comportamento do sistema com um passo a frente sem erros significativos. No entanto, estes resultados são dependentes do seu projeto, ou seja, números de entradas da RNA, números de neurônios na camada oculta, método de treinamento usado e também que dados são utilizados nas fases de estimação e validação do modelo neural.

## 7.2 Perspectivas futuras.

Espera-se que o conteúdo aqui exposto seja uma referência bastante útil para a solução de problemas de modelagem em sistemas dinâmicos em geral, ou até mesmo como material de consulta para o meio acadêmico para futuros projetos relacionados. Vale ressaltar que este trabalho não deseja ser a única solução para o problema apresentado.

Seguem abaixo algumas sugestões para futuras pesquisas:

- Utilização da estimação em uma gama de motores com potências e características diferentes, para confirmar a generalização dos resultados apresentados;

- comparação de outras técnicas de estimação, como por exemplo, RNA do tipo RBF, modelos autoregressivos, Algoritmos Genéticos, Filtros de Kalman entre outros;
- teste do método aplicado à estimação on-line, possibilitando o rastreamento em tempo real dos parâmetros do equipamento, importante para Controle Adaptativo;
- utilização dos parâmetros obtidos através da estimação on-line para um esquema de controle. De início, em condições normais de operação, que é conhecido como modo servo. Em seguida no modo regulatório, ou seja, na presença de distúrbios no sistema. Um exemplo de sistemas de controle é o Controle Adaptativo via RNA.

## Referências Bibliográficas

1. Action Technology (2011). [www.actiontechnology.com.br](http://www.actiontechnology.com.br). em 09/2011.
2. Aguirre, L. A. (2004). *Introdução a Identificação de Sistemas*. Belo Horizonte: UFMG, 2 edition.
3. Ballone, G. J. (2008). Neurônios e neurotransmissores. PsiqWeb, Internet, disponível em [www.psiqweb.med.br](http://www.psiqweb.med.br). Acessado em 05/2012.
4. Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (1993). *Nonlinear programming*. USA: John Wiley & Sons Canada, Ltd., 2 edition. ISBN-10: 0471557935.
5. Beale, M. H., Hagan, M. T., & Demuth, H. B. (2010). *Neural Network Toolbox™ 7*. USA: MathWorks, 7 edition. ISBN 0-9717321-0-8.
6. Beale, R. & Jackson, T. (1992). *Neural Computing: An Introduction*. Institute of Physics Publishing, Bristol, UK.
7. Braga, A. P., Carvalho, A. P. L. F., & Ludermir, T. B. (2000). *Redes Neurais Artificiais*. LTC, Rio de Janeiro.
8. Caner, E. S., Albuquerque, M. P., Albuquerque, M. P., & Gesualdi, A. R. (2004). *Comparações entre Redes MLP e RBF no Reconhecimento de Caracteres em Imagens de Placas de Automóveis*. Notas técnicas. Rio de Janeiro, RJ, Centro Brasileiro De Pesquisas Físicas - CBPF. CBPF-NT-008/04.
9. Carvalho, A. S., Braga, A. G., & Figueiredo, H. A. (2004). *Identificação por RNA da Relação Vazão de Alimentação por Temperatura de Topo de uma Coluna de Destilação Didática*. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Automação Industrial) - Campos dos Goytacazes, RJ, Centro Federal de Educação Tecnológica de Campos - CEFET Campos.

10. Cerqueira, J. J. F. (2001). *Identificação de sistemas dinâmicos usando Redes Neurais Artificiais: uma aplicação a manipuladores robóticos*. Tese (Doutorado em Engenharia Elétrica) - Campinas, SP, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas - UNICAMP.
11. Coelho, A. A. R. & Coelho, L. S. (2004). *Identificação de Sistemas Dinâmicos Lineares*. Editora UFSC.
12. Corrêa, G. R. (1995). *Desenvolvimento e implementação em hardware de um controlador neural*. Dissertação (Mestrado em Engenharia Elétrica) - Uberlândia, Departamento de Engenharia Elétrica, Centro de Ciências Exatas, Universidade Federal de Uberlândia.
13. Costa e Silva, C. A. M., Quintaes, B. R., Pastura, C. P., Henriques, A. L. M., & Santos, F. M. (2005). Caracterização microbiológica de resíduos sólidos domiciliares de bairros e comunidades carentes da cidade do rio de janeiro. Congresso Brasileiro de Engenharia Sanitária e Ambiental, 23. Campo Grande, MS, Associação Brasileira de Engenharia Sanitária e Ambiental - ABES.
14. Datta, B. N. (1995). *Numerical Linear Algebra and Applications*. Brooks, USA, 1995.
15. De Castro, R. E. (2001). *Otimização de estruturas com Multi-objetivos Via Algoritmos Genéticos de Pareto*. Tese (Doutorado em Engenharia Civil) - Rio de Janeiro, Universidade Federal de Rio de Janeiro / COPPE.
16. Dorf, R. C. & Bishop, R. H. (2004). *Modern Control Systems*. 10th ed., Hardcover.
17. Feedback Instruments Ltda. (1995a). *Analogue Servo - Fundamentals Trainer - 33-002*. Manual do usuário. Ed.03.
18. Feedback Instruments Ltda. (1995b). *Digital Servo - Fundamentals Trainer - 33-003 SFT154*. Manual do usuário. Ed.02.
19. Franklin, G., Powell, D., & Workman, M. (1990). *Digital Control of Dynamic Systems*. 3rd ed., Addison-Wesley.
20. Freeman, J. A. & Skapura, D. M. (1991). *Neural Network*. Addison Wesley Pub. Comp.

21. Furtado, H. C. M. (2008). *Redes neurais e diferentes métodos de assimilação de dados em dinâmica não linear*. Dissertação (Mestrado em Computação Aplicada)- São José dos Campos, Instituto Nacional de Pesquisas Espaciais - INPE, 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.
22. Haykin, S. (2001). *Redes Neurais: Princípios e práticas*. Bookman, Porto Alegre.
23. Juvinao Carbono, A. J. (2001). *Otimização da disposição de linhas de ancoragem utilizando Algoritmos Genéticos*. Tese (Doutorado em Engenharia Civil) - Rio de Janeiro, Departamento de Engenharia Civil PUC-Rio.
24. Kuga, H. K. (2005). *Noções práticas de técnicas de estimação*. Notas de aula. São José dos Campos, SP, Instituto Nacional de Pesquisas Espaciais - INPE.
25. Leite, A. C. (2007). *Detecção e diagnóstico de falhas em sensores e atuadores da plataforma multi-missão*. Dissertação (Mestrado em Engenharia e Tecnologia Espaciais / Mecânica Espacial e Controle) - São José dos Campos, Instituto Nacional de Pesquisas Espaciais - INPE, 372p. (INPE-15219-TDI/1313).
26. Leite, A. C., Kuga, H. K., & Lopes, R. V. F. (2006). Model-based fault detection on a reaction wheel. Brazilian Conference on Dynamics, Control and their Applications, 5. Guaratinguetá, SP, Sociedade Brasileira de Automática - SBA.
27. Leonhard, W. (2001). *Control of Electrical Drives*. 3rd ed., Springer Verlag, Berlin.
28. Maitelli, A. L. & Filho, O. G. (2003). Controlador híbrido indireto baseado em redes neurais - parte i: desenvolvimento e implementação. Simpósio Brasileiro de Automação Inteligente, VI. Bauru, SP.
29. Masters, T. (1994). *Signal and Image Processing With Neural Networks -AC++ Sourcebook*. 3rd ed., Jhon Wiley & Sons, Inc, New York, USA.
30. McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, vol 23(n 5), 115–133.
31. Meister, D. (2006). *Aplicação do Método dos Mínimos Quadrados na estimação de parâmetros do modelo de um transformador*. Dissertação (Mestrado em Enge-

- nharia Elétrica) - Brasília, DF, Departamento de Engenharia Elétrica, Universidade de Brasília, 115p. Publicação PPGENE.DM-284A/06.
32. Moreira, I. M. (2010). *Projeto e Implementação de um Controle de Velocidade Digital Auto-Ajustável para um Motor de Corrente Contínua*. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) - Campos do Goytacazes, RJ, Instituto Federal Fluminense - IFF.
  33. Navarro, C. M., Carvalho, A. S., & Paula Junior, G. G. (2010). Estimação de parâmetros de um motor cc por meio de redes neurais artificiais (rna). Simpósio Brasileiro de Pesquisa Operacional - SBPO, XLII. Bento Gonçalves - RS, Sociedade Brasileira de Pesquisa Operacional - SOBRAPO.
  34. Navarro, C. M. & Leite, A. C. (2010). Implementação digital de um controlador adaptativo por modelo de referência (mrac) para um motor cc. 14º Congresso Internacional e Exposição Sul-Americana de Automação, Sistemas e Instrumentação. Brazil Automation - ISA 2010. Associação Sul-Americana de Automação - ISA Distrito 4.
  35. Ogata, K. (2004). *Engenharia de Controle Moderno*. Rio de Janeiro: LTC, 4 edition.
  36. Oliveira, P. J. R. D. (1998). *Estimação de parâmetros do motor de indução utilizando um modelo Contínuo no tempo*. Dissertação (Mestrado em Engenharia Elétrica) - Belo Horizonte, MG, Universidade Federal de Minas Gerais - UFMG.
  37. Oliveira, R. H. (2004). *Estimação de parâmetros do motor de indução trifásico com o uso de redes neurais recorrentes*. Dissertação (Mestrado em Engenharia Elétrica) - Itajubá, MG, Universidade Federal de Itajubá - UNIFEI.
  38. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. Cornell Aeronautical Laboratory.
  39. Siemens (2006). *Motores de corrente contínua: Guia rápido para uma especificação precisa*. Unidade Automação e Controle - Acionamentos e Motores Elétricos. Publicação técnica. Ed.01.

40. Silva, H. G. D. & Otal, L. O. (2010). *Um estudo comparativo entre RNA's e Modelos Autoregressivos para identificação de uma coluna de destilação piloto*. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) - Campos do Goytacazes, RJ, Instituto Federal Fluminense - IFF.
41. Sousa, M. R. D. (2009). *Estimação utilizando mínimos quadrados recursivo e controle de velocidade do motor DC Servo Trainer*. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica)- Fortaleza, CE, Universidade Federal do Ceará - UFC, 63p.
42. Souza, M. J. F. (2003a). *Inteligência Computacional para Otimização*. Notas de aula (Departamento de Computação, Instituto de Ciências Exatas e Biológicas - ICEB) - Ouro Preto, MG, Universidade Federal de Ouro Preto - UFOP. <http://www.decom.ufop.br/prof/marcone>.
43. Souza, M. J. F. (2003b). *Métodos numéricos e estatísticos*. Notas de aula (Departamento de Computação, Instituto de Ciências Exatas e Biológicas - ICEB) - Ouro Preto, MG, Universidade Federal de Ouro Preto - UFOP. <http://www.decom.ufop.br/prof/marcone>.
44. Tinós, R. (1999). *Detecção e diagnóstico de falhas em robôs manipuladores via Redes Neurais Artificiais*. Dissertação (Mestrado em Engenharia Eletrica) - São Carlos, SP, Escola de Engenharia de São Carlos da Universidade de São Paulo - USP.
45. Vargas, J. A. R. (1997). *Identificação de sistemas dinâmicos via redes neurais artificiais*. Dissertação (Mestrado em Engenharia Eletrônica e Computação) - São José dos Campos, SP, Instituto Tecnológico de Aeronáutica - ITA. <http://www.bd.bibl.ita.br/tde-busca/arquivo.php?codArquivo=273>. 04, 2010.
46. WEG (2004). Motores de corrente contínua. *Catálogo*. Jaraguá do Sul, SC. 8p.
47. Widrow, B. & Hoff, M. E. (1960). Adaptive switching circuits. Western Electronic Show and Convention. Institute of Radio Engineers.

## Apêndice A

# INSTALAÇÃO DO KIT E COMUNICAÇÃO COM O MATLAB®

Antes de começar qualquer experimento, foi necessário investigar todo o procedimento de instalação do kit com o PC. Em seguida, as configurações de comunicação com o Matlab® também foram estudadas. O presente apêndice tem como objetivo registrar a forma de instalação e comunicação a fim de facilitar futuros trabalhos com o equipamento.

Primeiramente deve-se fazer a ligação da UM-33-100 com a fonte de alimentação elétrica. A UM-33-100 possui uma porta de comunicação paralela de 34 vias que serve para comunicação e com a UD-33-120, serve também para energizá-la.

A seção a seguir trata do procedimento de ligação da UD-33-120.

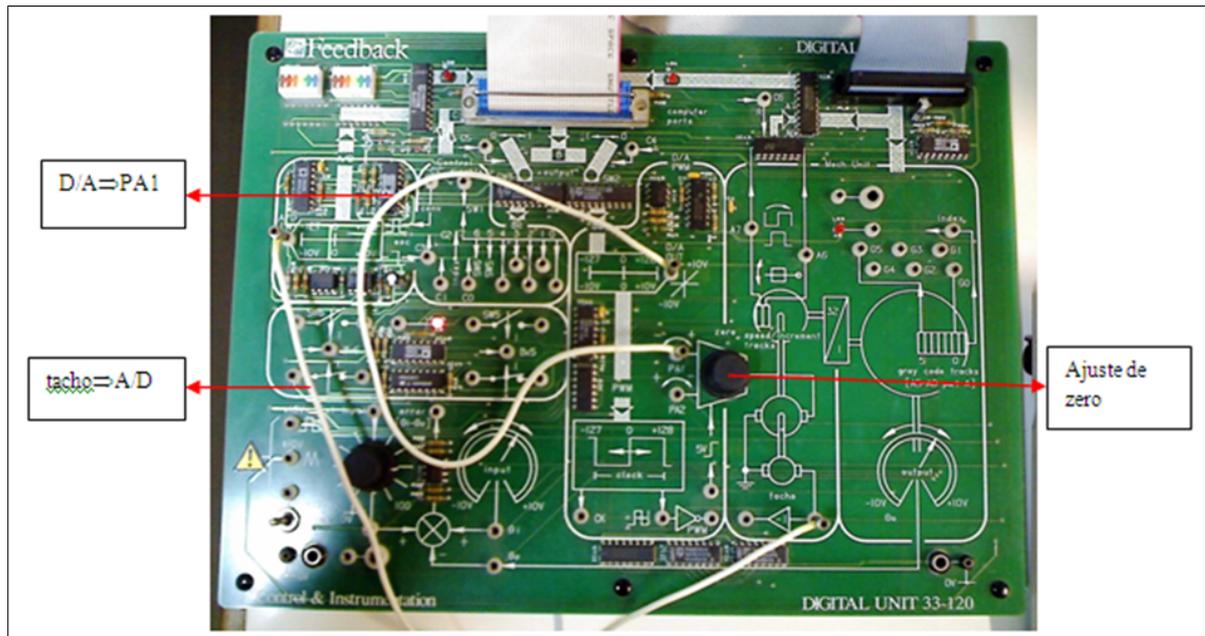
### A.1 Ligação da UD-33-120.

As conexões elétricas da UD-33-120 são feitas da seguinte maneira:

- Manter o kit desligado;
- Fazer a conexão do borne “tacho” até “A/D”. Neste caso, estaria habilitando o uso do sensor de velocidade (tacogerador). O conversor A/D recebe o sinal do sensor e converte para sinal digital antes de enviar para o PC;
- Fazer a conexão do borne “D/A” até o borne “PA1”. O borne “PA1” serve para enviar a informação elétrica diretamente para a UD-33-100. Este sinal pode ser proveniente do potenciômetro ou do PC após ser convertido em sinal analógico pelo conversor D/A;
- Ao ligar o kit, o motor cc ficará com uma velocidade residual. Deve-se, então,

fazer o ajuste de zero para que ele fique parado. Isso é feito através do potenciômetro destacado na figura A.1.

A ligação entre o kit e o PC é feita através de um cabo paralelo de 37 vias ligado a uma placa de aquisição de dados instalada no PC. A figura A.1 ilustra os procedimentos descritos.



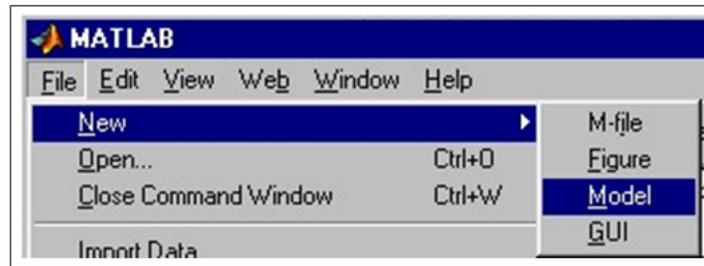
**Figura A.1** Conexões da UD-33-120.

A próxima seção é descrito todo o procedimento de configuração do Simulink para que o mesmo comunique corretamente com o kit.

## A.2 Configuração do Simulink para comunicar com o kit.

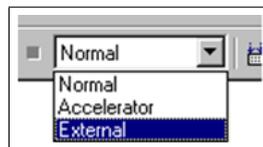
Para fazer as configurações no software é preciso ligar o PC e abrir o Matlab®. Em seguida, abrir o Simulink e executar os seguintes passos:

- “File”, “New”, “Model” para abrir um arquivo novo. Figura A.2:



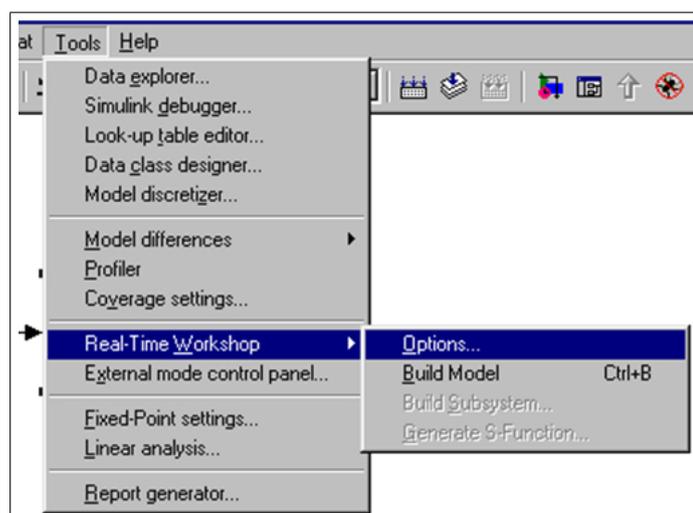
**Figura A.2** Abrindo um arquivo novo.

- Ir ao browser onde consta escrito “Normal” e mudar para “External”. Figura A.3:

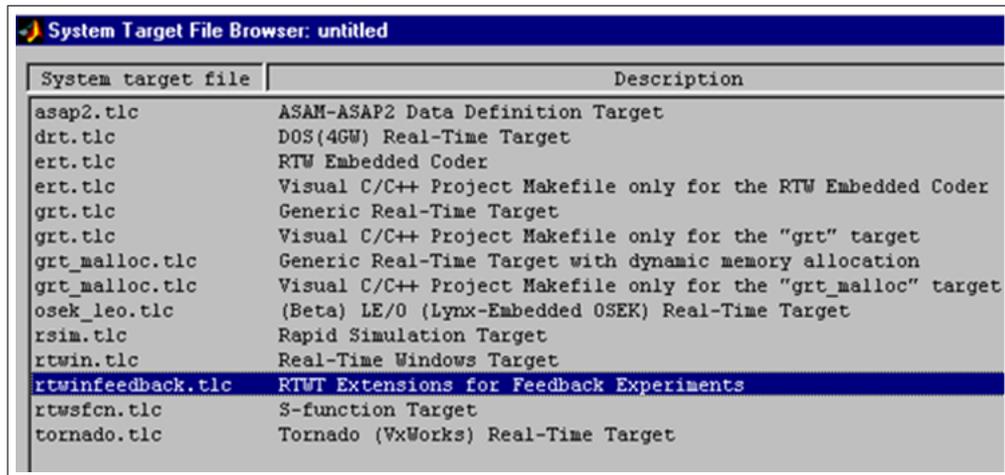


**Figura A.3** Modificando para simulação externa.

- Clicar em “Tools”, “Real-time Workshop”, “Options”, figura A.4:
- Vai abrir a janela igual a figura A.5, então selecionar: “rtwinfeedback.tlc RTWT Extensions for Feedback Experiments”, clicar em “OK”;

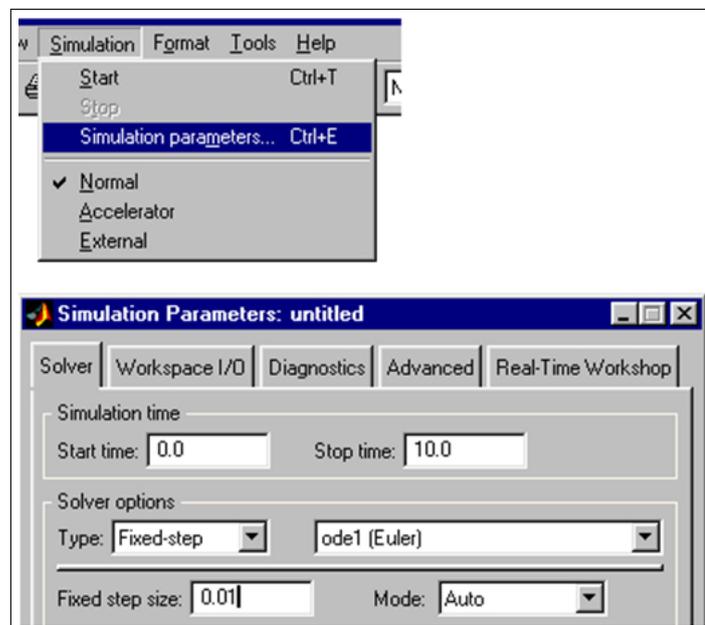


**Figura A.4** Acessando opções de comunicação.



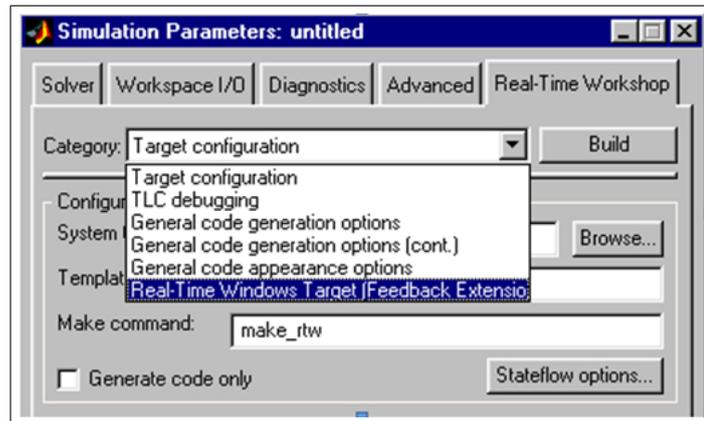
**Figura A.5** Selecionando drive do kit.

- Clicar em "Simulation", selecionar "Simulation parameters". Figura A.6:
- Na aba Solver (tempo de execução), ir em "solver options", modificar "Type = Fixed-step" (Integrador de passo fixo) e também "ode 1 (Euler)";
- Em seguida selecionar "Fixed-step size = 0.01" e "Mode = auto".



**Figura A.6** Configurando os parâmetros de simulação I.

- Na aba "Real-Time Workshop" ir em "Category", selecionar "Real-Time Targets [Feedback extension]", "OK". Veja figura A.7:



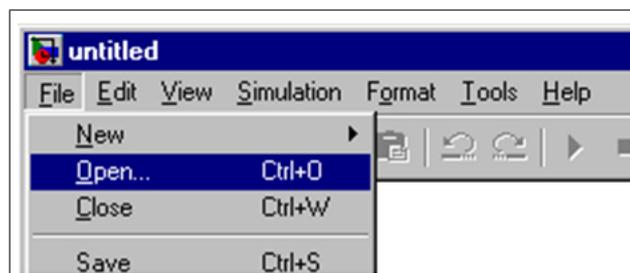
**Figura A.7** Configurando os parâmetros de simulação II.

Feitas as configurações no Simulink, a seção seguinte apresenta os passos para criação de diagramas de blocos para fazer simulações.

### A.3 Criação de diagrama de blocos.

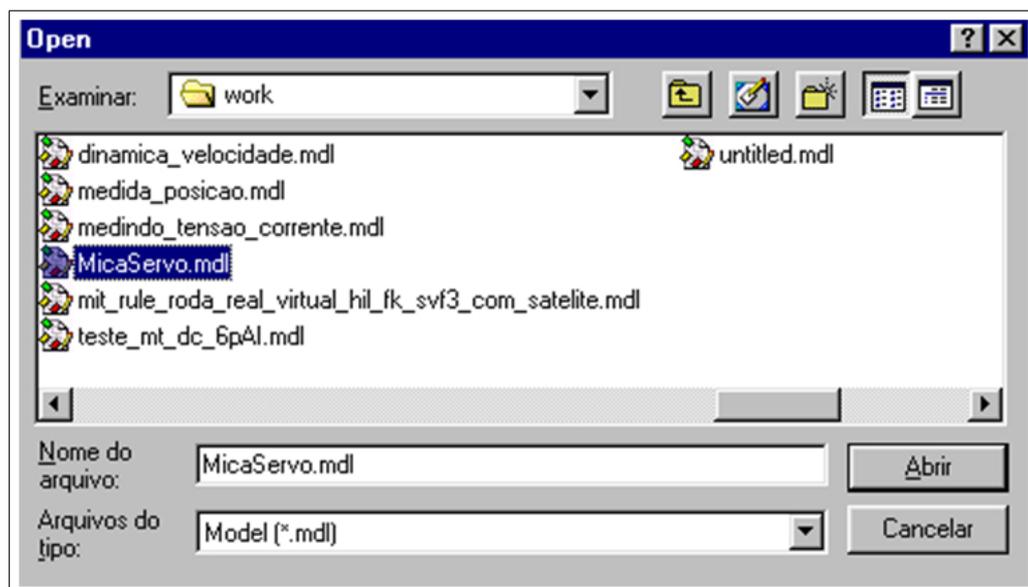
Para criar diagramas de blocos aplicados especificamente ao kit é preciso inserir os blocos A/D e D/A no arquivo gerado anteriormente. Os passos a seguir ilustram como é feito esse procedimento.

- No arquivo gerado, ir em “File”, “Open”, figura A.8.



**Figura A.8** Abrindo bloco de comunicação do kit I.

- Procurar por: “Mica Servo.mdl”. Figura A.9.



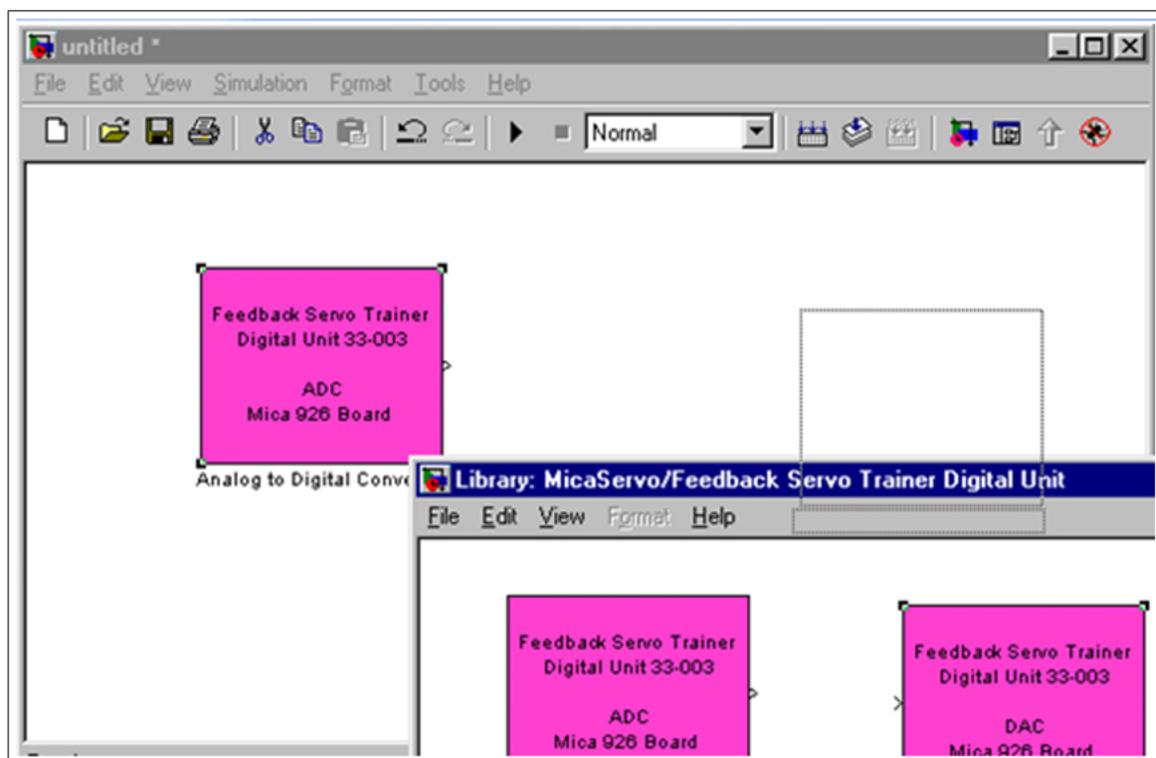
**Figura A.9** Abrindo bloco de comunicação do kit II.

- Dar dois cliques e entrar no bloco “Digital Servo Fundamentals with MICA 926” (Verde). Figura A.10.



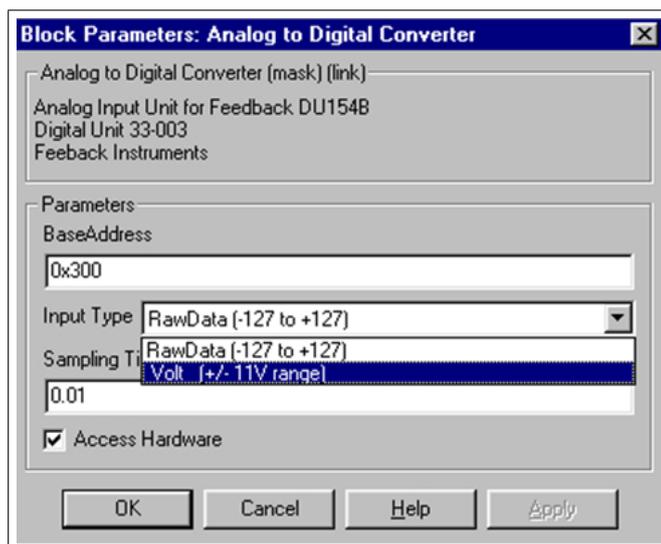
**Figura A.10** Abrindo bloco de comunicação do kit III.

- Selecionar e arrastar os blocos “ADC Mica 926 Board” e “DAC Mica 926 Board” (Rosas), figura A.11, para o arquivo que foi gerado. Esses blocos são os drives de comunicação do kit.



**Figura A.11** Inserindo os blocos dos conversores A/D e D/A no arquivo.

- Dois cliques no bloco “ADC Mica 926 Board” (Rosa), vai abrir uma janela igual a figura A.12. Selecionar em “Input type”: “+/- 11v range”;
- Digitar em “Sampling Time”: “0.01” (Refere-se ao menor período de amostragem que consegue alcançar), clicar “OK”;

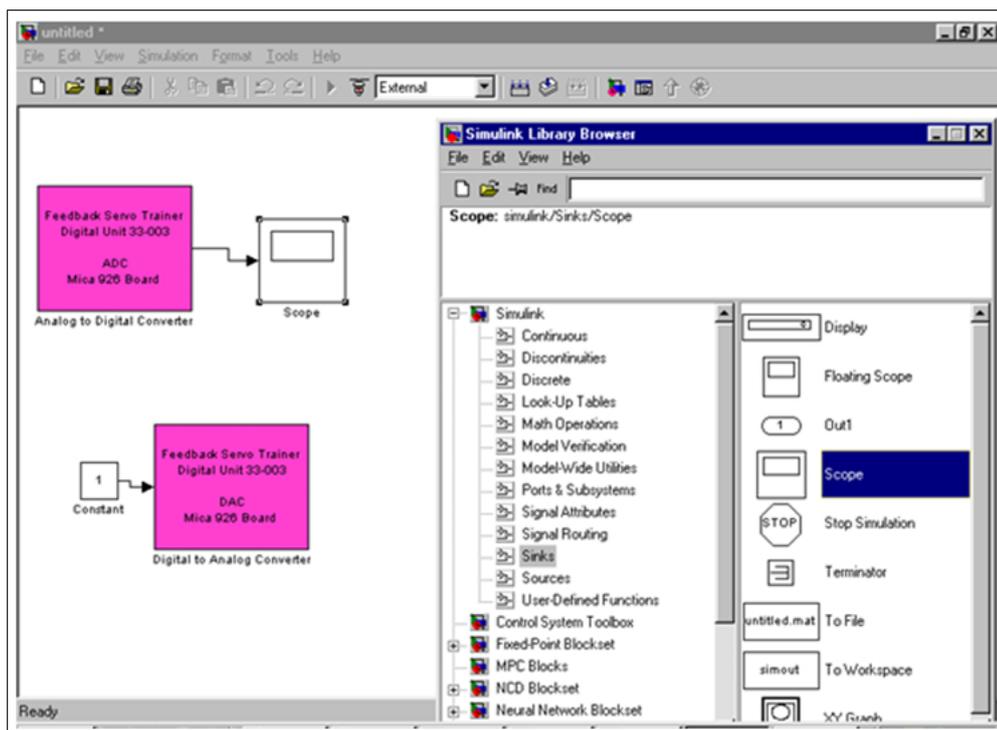


**Figura A.12** Configurando os blocos dos conversores A/D e D/A.

Repetir o mesmo procedimento no bloco “DAC Mica 926 Board”.

Com esses blocos (ADC e DAC), é possível enviar e receber informações do kit. Para fazer um teste e verificar se está comunicando corretamente os passos seguintes comenta o procedimento:

- Abrir o “Simulink Library Browser”, onde se encontra diversos blocos para se fazer a montagem dos diagramas. Ir a opção “Sources”, em seguida selecionar o bloco “Constant” e arrastar para o arquivo gerado;
- Na opção “Sinks” da biblioteca selecionar o bloco “Scope” e arrastar para o arquivo gerado. Conectar o bloco “Constant” no “DAC” e o “Scope” no “ADC” como mostrado na figura A.13.



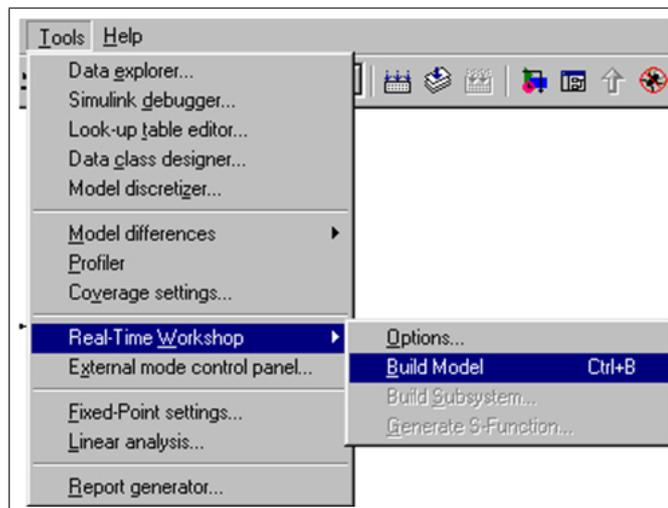
**Figura A.13** Montagem de um diagrama de blocos básico.

A figura A.13 é um exemplo bem simples de um digrama de blocos que interage com o kit. Com o bloco “Constant” é possível enviar valores para o equipamento. Com o bloco “Scope” é possível visualizar graficamente os valores de funcionamento do motor cc. O bloco “Scope” também envia informações numéricas para a área de trabalho do Matlab para possíveis análises da simulação.

Na próxima seção é comentado como é feita a comunicação do Software com o kit.

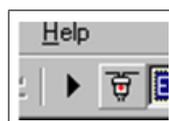
#### A.4 Comunicação.

Feita todas as configurações é preciso que o Matlab gere um código em "C" com todas as opções selecionadas. Para isso, basta clicar em "Tools", "Real-Time Workshop", selecionar "Build Model" (aguardar enquanto o MATLAB® gera o código em "C" e compila o arquivo .mdl). Figura A.14.



**Figura A.14** Gerando código "C" e compilando o arquivo.

Para a comunicação com o kit, basta agora clicar em "Connect to Target" e selecionar o "play". O primeiro passo é o botão da direita e o outro é o botão da esquerda como na figura A.15.



**Figura A.15** Dando inicio a comunicação.

Neste instante é preciso enfatizar que para qualquer mudança de estrutura no arquivo, haverá a necessidade de compilá-lo novamente.