

NOV
02.56
1925m
118/MEM

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA MÊCANICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
SISBI/UFU



1000220674

**METODOLOGIA DETECÇÃO E DIAGNOSTICO DE FALHAS EM
SISTEMAS DE REFRIGERAÇÃO USANDO REDES NEURAIS**

Dissertação apresentada

à Universidade Federal de Uberlândia por:

JOSÉ ANTONIO TUMIALÁN BORJA

Como parte dos requisitos para obtenção do título
de Mestre em Engenharia Mecânica.

Tendo como Banca Examinadora:

Prof. Dr. Oscar S. Hernandez Mendoza (UFU) - Orientador

Prof. Dr. Ricardo Nicolau Nassar Koury (UFMG)

Prof. Dr. José Francisco Ribeiro (UFU)

Uberlândia 13 de Setembro de 2002



ALUNO: José Antonio Tumialán Borja

NÚMERO DE MATRÍCULA: 5001606-4

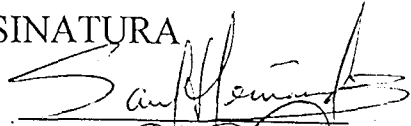

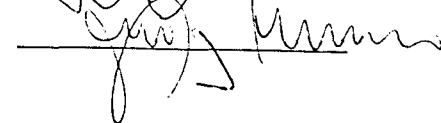
ÁREA DE CONCENTRAÇÃO: Transferência de Calor e Mecânica dos Fluidos
PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA: NÍVEL MESTRADO

TÍTULO DA DISSERTAÇÃO:

“Metodologia Detecção e Diagnóstico de Falhas em Sistemas de Refrigeração Usando Redes Neurais”

ORIENTADOR: Prof. Dr. Oscar Saul Hernandez Mendonza

A Dissertação foi **APROVADA** em reunião pública, realizada no Anfiteatro do Bloco X do Campus Santa Mônica, em 13 de setembro de 2002, às 13:00 horas, com a seguinte Banca Examinadora:

NOME	ASSINATURA
Oscar Saul Hernandez Mendonza, Prof. Dr.	UFU 
José Francisco Ribeiro, Prof. Dr	UFU 
Luiz Machado, Prof. Dr.	UFMG 

Uberlândia, 13 de setembro de 2002.

*A meus pais,
Hermenegilda e em memória Sebastián
Pela confiança ao longo dos anos.*

Meus Agradecimentos,

Ao Professor Oscar Saul Hernandez pela orientação, apoio durante o transcurso do curso, agradecimento especial ao colega do laboratório José Eduardo Oliveira pela ajuda do transcurso da dissertação.

À galera "Arrabal", pelo apoio e incentivos para continuar meus estudos.

Aos colegas do laboratório LEST: pelo companheirismo prestado.

À CAPES (Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) Pelo Apoio financeiro.

À EMBRACO, pela doação do equipamentos para a montagem da bancada de ensaios.

Ao técnico Cláudio Gomes do Nascimento, pela ajuda na montagem da bancada de ensaios.

METODOLOGIA DETECÇÃO E DIAGNOSTICO DE FALHAS EM SISTEMAS DE REFRIGERAÇÃO USANDO REDES NEURAIS

SUMÁRIO

Capítulo I		INTRODUÇÃO	1
	Aplicações em Sistemas HVAC		2
Capítulo II		BANCADA EXPERIMENTAL	6
2.1	Descrição do Protótipo de Sistemas de Refrigeração		6
2.2	Dimensionamento do Sistema de Refrigeração		7
	2.2.1 Seleção do Compressor		7
	2.2.2 Projeto do Condensador e Evaporador		10
	2.2.3 Válvula de Expansão		13
	2.2.4 Dispositivos Auxiliares		14
Capítulo III		INSTRUMENTAÇÃO E HARDWARE DE AQUISIÇÃO DE DADOS	15
3.1	Hardware e Filtragem de Sinais		15
3.2	Sistemas de Medição e Filtragem		16
	3.2.1 Fontes de Alimentação		16
	3.2.2 Instrumentação e Descrição dos Sensores		16
3.3	Hardware de Aquisição de Dados		20
3.4	Software de Monitoração		20
3.5	Curvas de Calibração		24
	3.5.1 Curvas de Calibração dos Transdutores de Pressão		24
	3.5.2 Curvas de Calibração dos Sensores de Temperatura		25
Capítulo IV		REDES NEURAIS ARTIFICIAIS- INSPIRAÇÃO BIOLÓGICA	28
4.1	Neurônio Artificial		29
4.2	Redes Neurais Artificiais		31

4.3	Treinamento das Redes Neurais Artificiais	32
4.4	Redes Backpropagation	32
4.4.1	Função de Ativação Para o Algoritmo Backpropagation	34
4.4.2	Implementação do Algoritmo de Propagação com a Função de Ativação	35
4.5	Metodologia Utilizada para o Treinamento da Rede Neural	40
Capítulo V	RESULTADOS EXPERIMENTAIS	41
5.1	Ciclo Padrão de Compressão a Vapor	41
5.2	Ciclo Real de Compressão a Vapor	42
5.2	Modelo de Detecção e Diagnostico de Falhas	43
5.4	Identificação Física dos Parâmetros	45
5.4.1	Sensibilidade de Parâmetros	47
5.4.2	Redução de Vazão de Água no Condensador	48
5.4.3	Redução de Vazão de Água no Evaporador	48
5.5	Estrutura das Redes Neurais	51
5.5.1	Treinamento das Redes Neurais para o Funcionamento Normal	51
5.6	Detecção de Falhas Utilizando esta Metodologia	53
5.6.1	Análise da Falha no Evaporador	54
5.6.2	Análise da Falha na Linha de Sucção do Compressor	57
5.6.3	Análise da Falha na Válvula de Expansão	59
Capítulo VI	CONCLUSÕES	63
Referências Bibliográficas		65
ANEXOS		
ANEXO I	68
ANEXO II	71

Simbologia

Letras Latinas

$APPR_c$	-	Diferença entre temperatura condensação e saída da água do condensador
$APPR_e$	-	Diferença entre temperatura de saída da água do evaporador e temperatura evaporação
C_p	-	Calor específico
COP	-	Coefficiente de performance
D	-	Diâmetro interno do tubo interno
$DTWCD$	-	Diferença de temperatura da água no condensador
$DTWEV$	-	Diferença de temperatura da água no evaporador
h_{lg}	-	Calor latente de vaporização
L	-	Comprimento da seção da tubulação
$LMTD$	-	Diferença média logarítmica da temperatura
l, g	-	Sub índice para líquido e vapor respetivamente
\dot{m}	-	Vazão mássica
P	-	Perímetro da tubulação
Pr	-	Número de Prandtl do refrigerante
\dot{Q}	-	Fluxo de transferência de calor
ref	-	Refrigerante
Re	-	Número de Reynolds do refrigerante
T_{sat}	-	Temperatura de saturação
$T_{ref,l}$	-	Temperatura de refrigerante em estado líquido
$T_{ref,su}$	-	Temperatura de entrada do refrigerante
$T_{ref,ex}$	-	Temperatura de saída do refrigerante
$T_{w,su}$	-	Temperatura de entrada da água

$T_{w,ex}$	-	Temperatura de saída da água
$T_{w,ex,ev}$	-	Temperatura de saída da água no evaporador
T_{ev}	-	Temperatura de evaporação
UA	-	Coefficiente global de transferência de calor
w	-	Água

Letras Gregas

α	-	Coefficiente de transferência de calor do refrigerante
μ	-	Coefficiente de atrito entre a came e o diafragma
μ	-	Função de pertinência
ρ	-	Densidade do refrigerante
λ	-	Condutividade térmica
ν_l	-	Viscosidade do refrigerante
κ_l	-	Condutividade do refrigerante

Tumialán, J. A, 2002, "Metodologia de Detecção e diagnóstico de falhas Usando Redes Neurais em Sistemas de Refrigeração", Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia, MG.

Resumo

Este trabalho apresenta uma metodologia de identificação de falhas, baseadas na análise de inter-relação de alguns parâmetros sensíveis para uma falha específica, imposta artificialmente em um protótipo de sistema de refrigeração, usando redes neurais artificiais (ANN). Esta metodologia é testada em um sistema de refrigeração de pequeno porte, as ANNs são treinadas inicialmente em operação normal do sistema (sem-falhas). Foram introduzidas três falhas artificiais ao sistema de refrigeração: redução da vazão de água do evaporador, fechamento brusco na válvula de expansão e obstrução na linha de sucção do compressor. A ANNs compara a operação normal e defeituosa, usando a análise do erro residuais na maioria dos parâmetros sensíveis, para detectar a falha imposta. Também é apresentado o método de treinamento empregado na estrutura básica da lógica usada para operar o método de detecção e diagnóstico da falha (FDD) proposto. A metodologia proposta mostrou-se bastante robusta, conseguindo identificar com bom nível de precisão a falhas introduzida ao sistema.

Palavra Chave: Detecção de falhas, Diagnóstico de falhas, Sistemas de Refrigeração, Redes Neurais.

Tumialán, J. A, 2002, "Fault Detection in Diagnosis Methodology Using Neural Networks in Refrigeration Systems", M. Sc. Dissertation, Universidade Federal de Uberlândia, MG

Abstract

This work presents a methodology of failure identification based on the inter-relation analysis of some sensitive parameters to specific fails artificially imposed on a prototype refrigeration system using Artificial Neural Network (ANN) modeling. This methodology is tested in a small refrigeration system, the ANN are initially trained in a normal operation of the system (without faults). Were introduces three artificial faults on the refrigeration system: reduction of evaporator water flow, suddenly expansion valve closure and obstruction on the compressor suction line. The ANN compares the normal and failure operation using a residual error analysis in the majority of two sensitive parameters to detect an imposed fault. Also, is presented the training method used in the basic structure of the logic used to operate the proposed detection and failure diagnostic method (DFD). The methodology proposed is robust, achieving a good precision level in the identification of faults introduced to the system.

Key words: Fault detection, Fault diagnosis, Refrigeration Systems, Neural Networks.

CAPITULO I

Introdução

A existência de falhas em sistemas de aquecimento, ventilação, ar condicionado e refrigeração (HVAC&R) possui um papel significativo na degradação dos níveis de conforto para os usuários destes sistemas. Os sistemas de refrigeração e ar condicionado participam do consumo total de energia nos setores residencial, comercial e industrial Brasileiro com 32 %, 20 % e 6 % respectivamente (Procel, 1998). Além disso, esta porcentagem pode ser maior em sistemas que operam com equipamentos defeituosos (Herzog e LaVine 1992). Com o avanço da tecnologia computacional, sistemas de aquisição de dados (A/D) e instrumentação, nos últimos anos fizeram possível utilizar algoritmos de monitoração em tempo real, que permitem a detecção de falhas que podem estar ocorrendo num sistema supervisionado e com isso impedir danos posteriores no sistema. Grande número de pesquisas para detectar e diagnosticar falhas (FDD) em sistemas de aquecimento, ventilação e ar condicionado (HVAC) são motivadas com o objetivo de reduzir o consumo de energia e aumentar conforto, para com isso obter uma significativa melhora dos níveis de vida.

Os sistemas HVAC foram abordados profundamente durante o decorrer das últimas décadas. Relatórios e documentos de controle destes sistemas são encontrados na literatura antes dos anos 70. Entretanto, suspeita-se fortemente que a crise de energia de 1970 influenciou a proliferação das pesquisas. Em 1974, a Agência Internacional de Energia (IEA) foi criada dentro dos padrões da organização para a cooperação econômica e o desenvolvimento (OECD) para elaborar um programa internacional de energia. Desde então, o IEA participou de muitos projetos de pesquisa, desenvolvimento e de demonstração (D&D) para prever exatamente o uso da energia nos edifícios, por meio de comparações com programas de computador existentes (monitorações de edifícios, comparação de métodos de cálculo, assim como a qualidade do ar e os estudos aprofundados sobre o tema propostos por IEA-1996).

Os grandes avanços foram a implementação de sistemas de inteligência artificial, que operam do mesmo modo que o cérebro humano e que são capazes de construir regras de controle baseado em experiências prévias.

Existem três tipos de sistemas de inteligência artificial (IA): sistemas especialistas, redes neurais artificiais e lógicas difusa. Os sistemas especialistas são sistemas que trabalham analisando uma base de dados, gerada pelos operadores, técnicos e engenheiros experientes. Os processos são analisados em tempo real e se avaliam os problemas de operação aplicando

regras do tipo *if/then/else*. Quando os parâmetros do sistema real variam, uma probabilidade de ocorrência é calculada. Se a probabilidade é baixa, o sistema especialista continua procurando outros motivos. Uma vez que o sistema só detecta uma causa quando a probabilidade é grande toma-se uma ação corretiva. O sistema armazena a causa com alta probabilidade e se a variação ocorrer novamente, o sistema especialista interage com o operador diretamente.

O segundo tipo de IA são as redes neurais. As redes neurais não dependem do conhecimento do especialista. Os dados disponíveis no processo são incorporados a algoritmos numéricos, para solucionar os problemas complexos baseados em aprendizagem por reconhecimento de um modelo. As redes neurais não precisam de iteração com o operador.

O terceiro tipo de IA, e bastante conhecido na literatura especializada de controle, são os sistemas baseados na lógica difusa, que também tomam decisões baseados em regras obtidas a partir de conhecimento prévio do sistema. Mais genérica que o sistema especialista, a lógica difusa é definida incorporando declarações do tipo *mais* ou *menos* nas regras de controle. Estas, por sua vez, são baseadas na instrução e experiência de um operador humano.

Aplicações em Sistemas HVAC

Nesta seção, apresenta-se as investigações de FDD (Fault Detection and Diagnosis), que envolvem aplicação de métodos específicos em subsistemas particulares. Os subsistemas descritos são sistemas de aquecimento, refrigeração, unidades de manejo de ar VAV e sistemas térmicos de armazenamento.

Sistemas de Refrigeração (SR)

Diversos estudos de investigação sobre FDD em subsistemas de refrigeração e/ou em seus componentes foram realizados. A seguir apresentaremos alguns aportes mais importantes.

(Braun e Rossi 1997) utilizaram os métodos da detecção automatizada das falhas em SR de compressão de vapor. (Braun e Brueker 1997) utilizaram uma metodologia estatística, baseada em regras definidas para unidades de condicionamento de ar de terraço (tipo "Split"). (Grimmelius et al. 1995) geraram uma matriz de sintomas para cada tipo de falha do SR e o

efeito causado por cada falha foi analisado. Os sintomas foram desvios de variáveis selecionadas de valores previstos, usando um modelo estatístico não linear. (Dexter e Mok, 1992) usaram um classificador difuso para identificar determinados padrões de falha, utilizando um modelo qualitativo.

O sistema de FDD para SR (Stylianou e Nikanpour 1996) contém três módulos básicos de detecção das falhas do SR. O primeiro módulo (ciclo-desligado) é realizado monitorando o desempenho de sensores que são selecionados quando o SR é desligado à noite. O segundo módulo detecta falhas durante a partida (módulo de arranque: primeiros 15 minutos) e trata daquelas relacionadas às características do fluxo do refrigerante. O último módulo (módulo de estado estacionário) detecta o declínio no desempenho quando a unidade está operando em uma condição do estado estável.

Unidade de Manejo de Ar (AHU)

O diagrama esquemático da unidade de manejo (AHU) de volume de ar variável (VAV) utilizado por (Lee, 1996) consiste de ventiladores, amortecedores, serpentina de resfriamento, sensores e controladores como mostra a Figura 1.

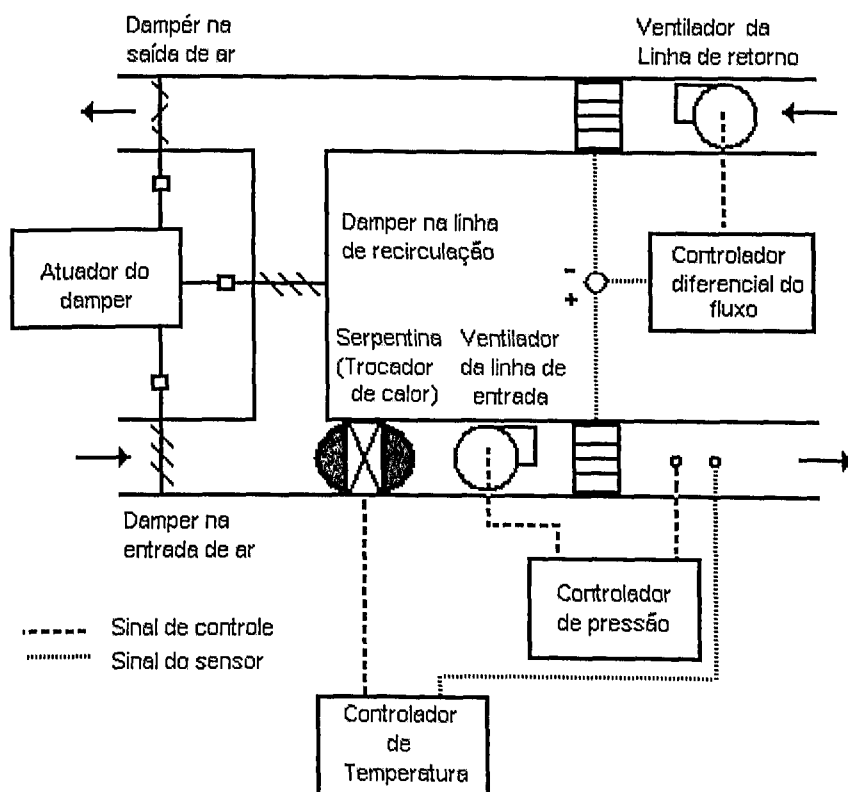


Figura 1. Diagrama esquemático da unidade de manejo de ar de volume de ar variável.

Quatro estudos consecutivos com métodos de FDD em VAV AHU foram executados: (Lee, 1996a), (Lee, 1996b), (Lee, 1997) e (Lee, 1999).

(Lee., 1996a) define as variáveis residuais que fornecem a diferença medida entre o estado de funcionamento normal (linha-base) do sistema e o estado atual. A falha é detectada pelos resíduos significativamente diferentes de zero. No segundo estudo (Lee, 1996b) uma única rede neural artificial (SANN), foi treinada com testes padrões residuais para diagnosticar falhas em vários subsistemas de AHU. O terceiro (Lee, 1997) descreve o uso de duas-etapas da rede neural artificial (ANN) para o diagnóstico em uma AHU simulada. A primeira etapa, a ANN é treinada para identificar o subsistema quando a falha ocorre. A segunda etapa, a ANN é treinada para o diagnóstico da causa específica da falha. As equações da regressão foram usadas para a recuperação de falhas dos sensores de temperatura. Este último trabalho foi realizado aproveitando algumas falhas encontradas no estudo anterior para não se ter a necessidade treinar novamente a ANN. (Lee, 1999) mostra a aplicação de técnicas e classificação rigorosa de FDD em VAV AHU.

Sistemas de Aquecimento (SA)

(Li, 1997) analisou o uso de um protótipo para FDD em um SA complexo. Primeiramente, uma base de dados foi gerada mediante a simulação do sistema, analisando o processo dinâmico de um modelo matemático de talhado do sistema de controle. O pré-processamento destes dados gerou sete deslocamentos predeterminados do desempenho (estimativas de parâmetros) que foram escolhidos arbitrariamente. Neste trabalho, utilizou-se um modelo caixa-preta com o formato de uma rede neural artificial (ANN), que foi treinada para representar um sistema de aquecimento, de tal forma que mais tarde esta ANN pudesse ser usada para analisar outros sistemas de aquecimento sem a necessidade de treinar novamente a ANN. Dois tipos de ANNs ([M] ANNs múltiplos e simples [S] ANNs) foram comparados detectando e diagnosticando seis tipos diferentes de falhas. O SANN apresentou melhores resultados. Um classificador baseado em conhecimento foi usado e utilizou um método probabilístico de reconhecimento do modelo.

Sistemas de Armazenamento Térmicos

O Software HVACSIM+ foi usado para simular as condições defeituosas do sistemas de armazenamento térmicos (Niwa e Nakahara 1997). As falhas simuladas neste estudo relacionam-se ao armazenamento do sistema HVAC, à planta e à habitação. Muitos diagramas foram gerados para as taxas de fluxo de massa, temperaturas de refrigeração dentro do

sistema de armazenamento antes da introdução das falhas. Mostram que as falhas simuladas, nos vários subsistemas tiveram um efeito claro nestes diagramas depois que as falhas foram introduzidas. O modelo de lógica difusa que é um tipo de raciocínio que define um conjunto de causas que explicam um conjunto de eventos (sintomas da falha) foi usado para detectar e diagnosticar falhas (Yamada e Kamimura 1996). Um sistema de armazenamento térmico foi usado como um protótipo de prova.

Este trabalho apresenta uma metodologia robusta de modelo experto de detecção e diagnóstico de falhas (FDD), testado e treinado num sistema protótipo de refrigeração de pequeno porte, utilizando redes neurais artificiais (ANNs) para modelar o sistema. A metodologia de FDD usada é baseada na análise de sensibilidade dos parâmetros obtidos por meio da comparação de uma operação normal (sem-falhas) com uma operação com falhas artificialmente introduzidas. São utilizadas as técnicas de erros residuais e análises estatísticas de evolução de parâmetros para diagnosticar as falhas. O trabalho também descreve a metodologia utilizada para treinar as ANNs e a estrutura lógica básica de operação do modelo experto.

Para atender estes objetivos, o trabalho está assim organizado: no capítulo II, apresenta-se o projeto e construção dos principais componentes da bancada de ensaios (sistema de refrigeração de pequeno porte). O capítulo III descreve a instrumentação utilizada na bancada experimental e uma breve descrição do hardware de filtragem de sinais, sensores utilizados e conclui com a especificação do sistema de aquisição e monitoramento de dados. O capítulo IV apresenta o conceito básico das redes neurais artificiais, descrevendo-se detalhadamente as ferramentas matemáticas básicas usadas para a implementação do código computacional. O capítulo V apresenta o resumo teórico do ciclo de refrigeração por compressão a vapor, o modelo de detecção de falha, identificação e análise de sensibilidade dos parâmetros para uma falha específica, resultados do treinamento da ANN para funcionamento normal e defeituoso e no final detecção e diagnóstico de falha usando esta metodologia. O Capítulo VI apresenta as conclusões e os futuros desdobramentos deste trabalho.

CAPÍTULO II

Bancada experimental

Neste capítulo será apresentado o projeto e detalhes da construção dos principais componentes da bancada de ensaios. A bancada é um protótipo de sistema de refrigeração por compressão de vapor de pequeno porte.

2.1 Descrição do protótipo de sistema de refrigeração

A bancada de ensaios mostrada na Figura 2.1 consiste em um protótipo de sistema de refrigeração por compressão de vapor, tendo como fluido de trabalho (primário) o refrigerante R134a e fluido secundário água que alimenta o condensador e o evaporador individualmente. O sistema é composto basicamente por: compressor hermético (~300W) de velocidade variável, condensador, evaporador do tipo anular operando em contra corrente, e uma válvula de expansão eletrônica. O sistema foi totalmente instrumentado com sensores de temperatura tipo PT-100 e sensores de pressão manométrica piezos-resistivos.

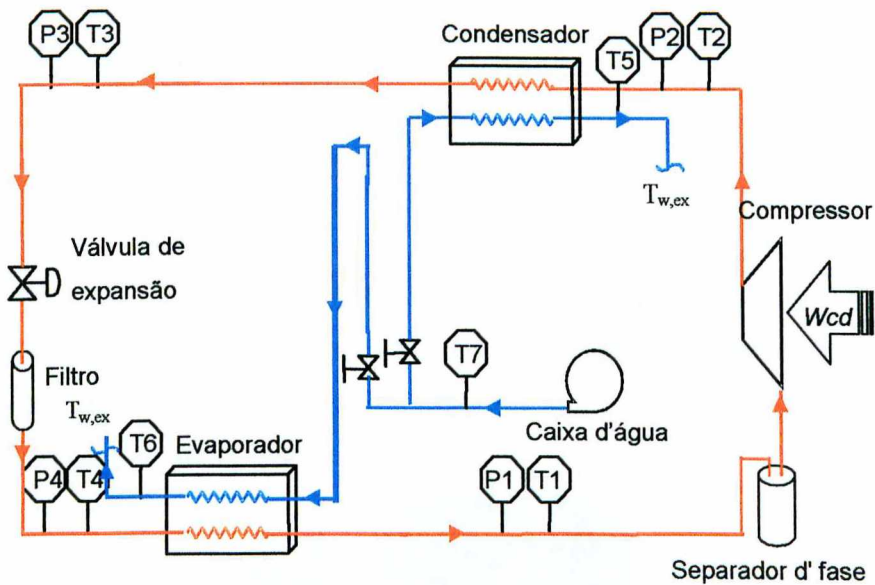


Figura 2.1 Representação gráfica do protótipo sistema de refrigeração.

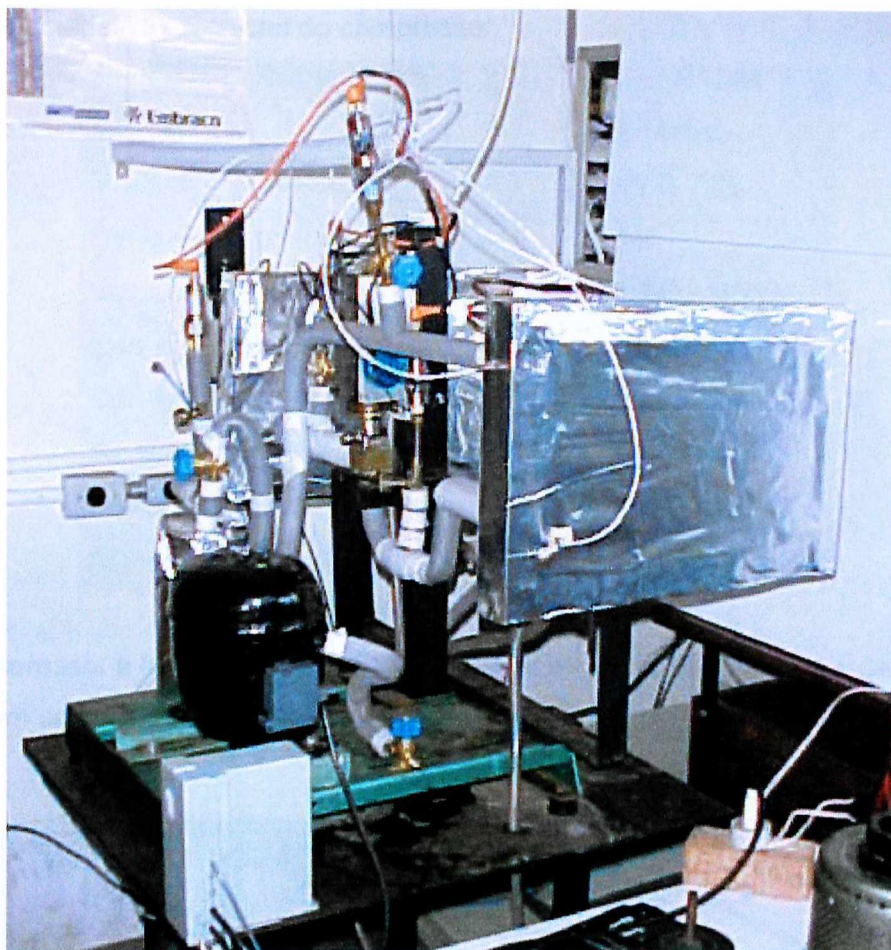


Figura 2.2. Visão geral do protótipo do sistema refrigeração montado no laboratório.

2.2 Dimensionamento do sistema de refrigeração

2.2.1 Seleção do compressor

O compressor utilizado é o compressor de velocidade variável VCC[®] Figura 2.2, desenvolvido pela Embraco S.A., que regula o consumo de eletricidade do motor elétrico de acordo com o esforço exigido. É do tipo alternativo, hermético, de um cilindro e suas especificações são apresentadas na Tabela 2.1



Figura 2.3 Compressor de capacidade variável VCC[®]

Tabela 2.1 Características técnicas do compressor.

Refrigerante	R134a
Marca	Embraco
Modelo	VEG TH8
Designação comercial [hp]	1/4
Velocidade de rotação [RPM]	1600 – 4500
Capacidade de refrigeração [Watt]*	80 – 770
Consumo de potência [Watt]*	60 - 330
Consumo de corrente [Ampere]*	0.45 – 2.2
Eficiência [Watt/Watt]*	0.27 – 2.52
Alimentação [Volts/Hz]	220V/60Hz

O compressor é ligado a uma rede monofásica através de um inversor de frequência, com esquema de ligação é ilustrado na Figura 2.4.

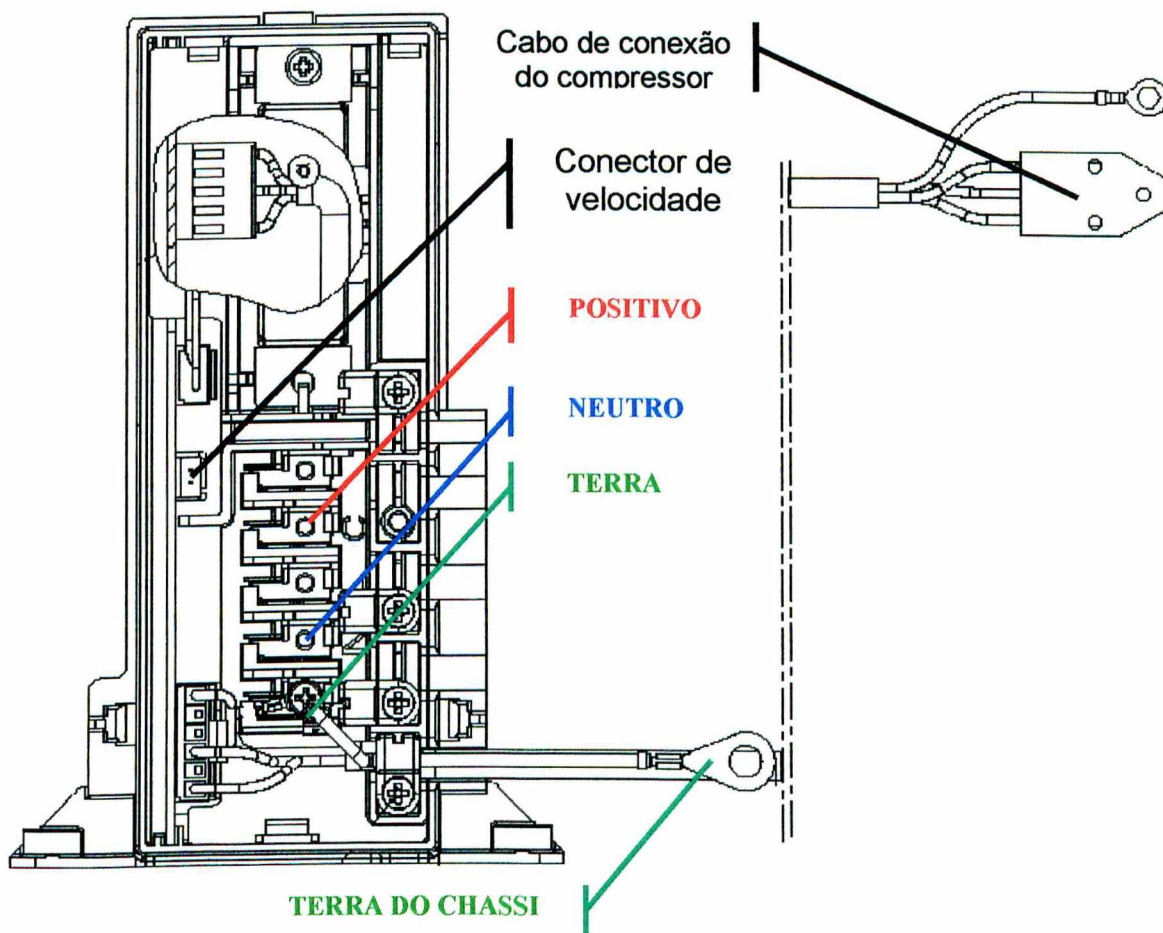


Figura 2.4 Inversor de frequência, esquema de ligação.

O controle da variação da velocidade no compressor é feito enviando um sinal ao inversor, que por sua vez aciona o compressor. O sinal enviado é um trem de pulsos contínuos, com amplitude de 0-5V como mostra a Figura 2.5

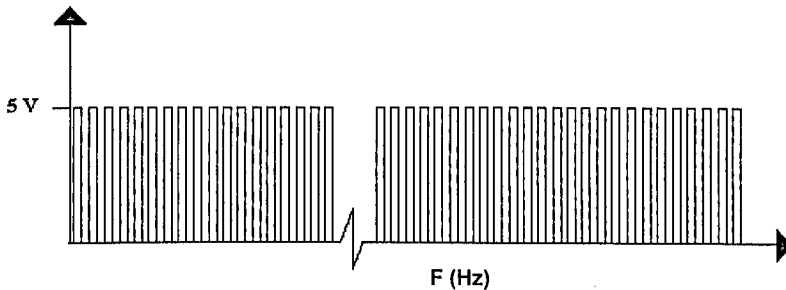


Figura 2.5. Diagrama do trem de pulsos.

O sinal é gerado por um microcomputador e enviado através da porta paralela para o inversor. A velocidade do compressor depende da frequência dos pulsos como mostra a Figura 2.6

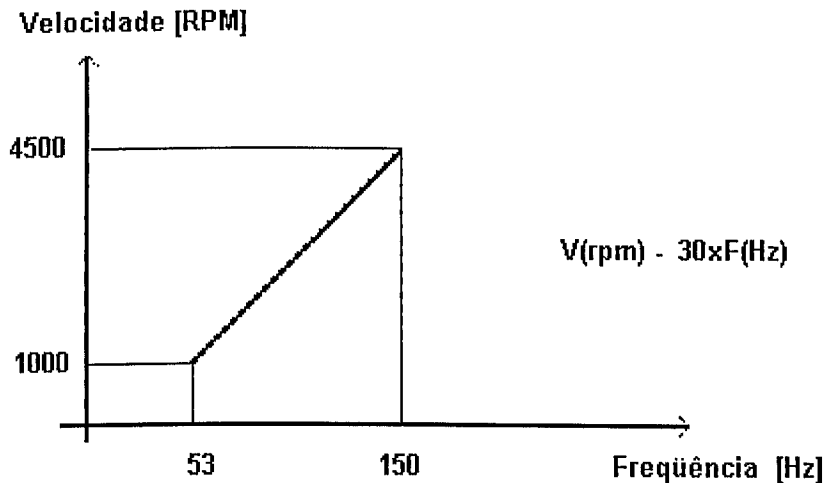


Figura 2.6. Diagrama Frequência x Rotações por minuto.

O dimensionamento dos demais componentes do sistema foi feito com base nas condições máximas de operação, recomendada pelos fabricantes, e a escolha do refrigerante R134a foi realizada seguindo os requisitos ambientais estipulados no protocolo de Montreal (1987), onde foram estabelecidas metas para a substituição dos clorofluorcarbonos (CFCs) por gases alternativos, menos prejudiciais ou inofensivos ao meio ambiente.

A Figura 2.7 apresenta um diagrama P x h com uma representação próxima da que seria o ciclo real de refrigeração por compressão de vapor, realizado pelo protótipo experimental.

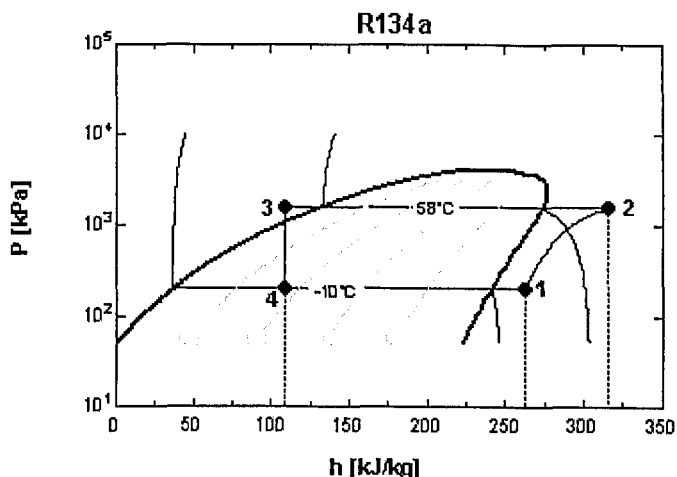


Figura 2.7 Ciclo teórico de compressão de vapor para o refrigerante R134a.

2.2.2 Projeto do condensador e evaporador

A seguir é apresentada uma breve descrição dos trocadores de calor. Um estudo mais detalhado foi apresentado por (Carvajal, Dissertação Mestrado, UFU, 2000).

O condensador consiste de dois tubos concêntricos onde o refrigerante escoa pelo tubo interno e a água pela parte anular entre os tubos, em uma direção de contrafluxo, conforme mostra a Figura 2.8.

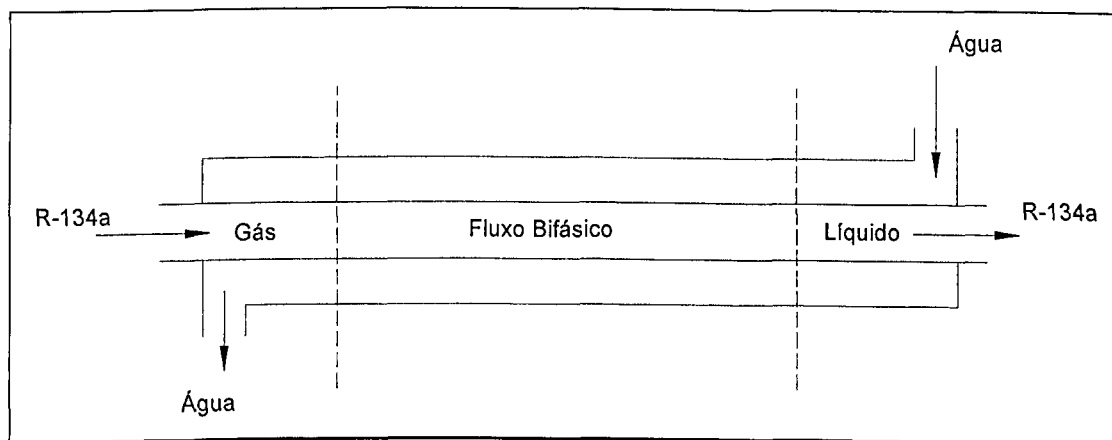


Figura 2.8 Representação esquemática do condensador.

Para atenuar as perdas para o meio ambiente, os tubos de ambos trocadores foram termicamente isolados com placas planas de lã de vidro protegidas com um revestimento externo de folhas de alumínio adesivo.

A Figura 2.9 mostra um trocador termicamente isolado montado na bancada de ensaios.

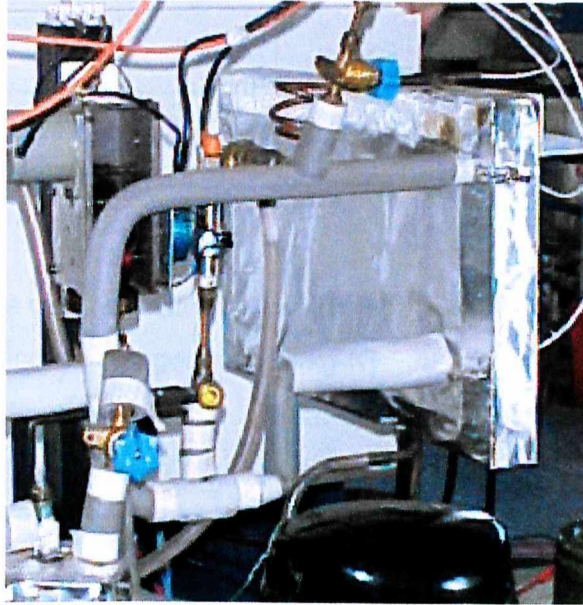


Figura 2.9 Foto do trocador de calor montado no protótipo do sistema refrigeração.

O condensador e evaporador foram dimensionados, utilizando diferentes correlações para obter os três coeficientes de troca de calor do refrigerante (gás, líquido e bifásico) e da água.

O coeficiente médio de transferência de calor por convecção do refrigerante nas fases gasosa e líquida foi obtido através da correlação de Dittus-Boelter, para transmissão de calor por convecção forçada no interior dos dutos, em que haja líquido e gases em escoamento turbulento:

$$\alpha = 0.023 \text{Re}^{0.8} \text{Pr}^{0.4} \left[\frac{\lambda}{D} \right] \quad (2.1)$$

onde:

- Re : N° de Reynolds
- Pr : N° de Prandtl
- λ : Condutividade térmica [kw/h m² °C]
- D : Diâmetro do tubo interno [m]

O coeficiente de transferência de calor do refrigerante no fluxo bifásico, foi determinado utilizando-se a correlação de Ackers e Rosson (ASHRAE Handbook of Fundamentals, 1993)

$$\alpha = 13.8 \left(C_p \frac{\nu_l}{k_l} \right)^{\frac{1}{3}} \left(\frac{h_{lg}}{0.75 C_p (T_{sat} - T_{ref,l})} \right)^{\frac{1}{6}} \left(D \frac{V_m}{\nu_l} \left(\frac{\rho_l}{\rho_g} \right)^{\frac{1}{2}} \right)^{\frac{1}{5}} \quad (2.2)$$

Onde:

- α : Coeficiente de troca de calor (Kw/m²-hr- °C)
- C_p : Calor específico do refrigerante [kJ/kg-K]
- ν_l : Viscosidade do refrigerante [kg/m-s]
- k_l : Condutividade do refrigerante [W/m-K]
- h_{lg} : Calor latente de vaporização
- T_{sat} : Temperatura de saturação [K]
- $T_{ref,l}$: Temperatura de refrigerante em estado líquido [K]
- D : Diâmetro da tubulação
- V_m : Velocidade média
- ρ : Densidade do refrigerante
- l, g : Sub índice para líquido e vapor respectivamente

Onde a equação (2.2) é válida somente para a faixa: $Re < 5000$ e $1000 < Re \frac{\rho_l}{\rho_g} < 20000$, segundo (ASHRAE Handbook, 1993).

Utilizando a correlação de Dittus-Boelter, foi calculado o coeficiente transferência de calor da água. Inicialmente foi suposto um valor inicial para vazão mássica de água igual a 2.8 [kg/min].

Através da conservação de energia obtêm-se as equações para o fluxo de calor para cada região do condensador.

$$\dot{Q} = \dot{m}_w C_{p_w} (T_{ref,su} - T_{ref,ex}) \quad (2.3)$$

$$\dot{Q} = \dot{m}_{ref} C_{p_{ref}} (T_{w,ex} - T_{w,su}) \quad (2.4)$$

$$\dot{Q} = UPL * LMTD \quad (2.5)$$

$$LMTD = \frac{(T_{ref,su} - T_{w,ex}) - (T_{ref,ex} - T_{w,su})}{\ln \left(\frac{(T_{ref,su} - T_{w,ex})}{(T_{ref,ex} - T_{w,su})} \right)} \quad (2.6)$$

Onde:

\dot{Q} :	Fluxo de calor	[kW]
\dot{m} :	Fluxo mássico	[kg/s]
C_p :	Calor específico	[kJ/kg-K]
T :	Temperatura	[K]
U :	Coeficiente global de transferencia de calor	[kW/m ² .K]
P :	Perímetro da tubulação	[m]
L :	Comprimento da seção da tubulação	[m]
$LMTD$:	Diferença média logarítmica da temperatura.	

Sub índices:

w : Água	ref : Refrigerante
su : Entrada	ex : Saída.

As equações 2.3, 2.4 e 2.5 são resolvidas para cálculo das seguintes incógnitas: \dot{Q} , $T_{w,ex}$ e L . O valor de L é obtido através da soma dos calores nas três regiões (Figura 2.8) este valor é igual ao calor total trocado no condensador. Carvajal et al, 2000 em seus cálculos encontrou um valor para o comprimento total dos tubos, tanto para o condensador como para o evaporador de 0.475 [m] para a vazão de água inicialmente definida.

2.2.3 Válvula de expansão

A válvula de expansão foi instalada entre o condensador e o evaporador, com o objetivo de promover a queda de pressão e temperatura isoentalpica entre os dois componentes. O líquido refrigerante vindo do condensador entra na válvula de expansão a alta pressão (pressão de condensação) e a deixa a baixa pressão (pressão de evaporação).

As válvulas de expansão podem ser: manuais ou automáticas. As válvulas de expansão manuais são aquelas em que a seção de fluxo do refrigerante permanece constante durante o processo. Já as válvulas de expansão automáticas são aquelas em que a seção de fluxo é alterada de acordo com as necessidades de trabalho. Válvulas de expansão termostáticas o seu funcionamento depende, tanto da pressão do evaporador quanto da pressão de comando do bulbo termostático.

Foi utilizado na bancada o mesmo dispositivo de expansão eletrônico utilizado por Carvajal 2000, trata-se de uma válvula de expansão termostática (TEV) modificada para ser

controlada eletronicamente através do motor de passo. Foi idealizada a partir de uma válvula termostática convencional. Com relação à configuração convencional, retirou-se da válvula a câmara do bulbo sensor, sendo substituída por um mecanismo de atuação constituído por um motor de passos, uma caixa redutora com quatro trens de engrenagens e um came que realiza a abertura ou fechamento da válvula.

Neste trabalho, a válvula não foi operada eletronicamente, optou-se pela operação manual da válvula durante a realização de todos os testes experimentais.

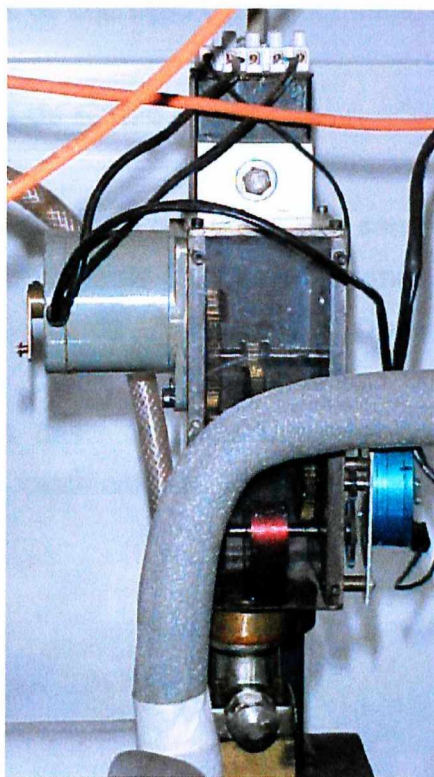


Figura 2.11. Foto do mecanismo de atuação da válvula de expansão.

2.2.4 Dispositivos auxiliares

Entre o condensador e a válvula de expansão foi instalado um filtro com a finalidade de garantir apenas a entrada de líquido sub-resfriado na entrada da válvula de expansão.

Para garantir que na sucção do compressor o refrigerante se encontre na fase vapor, foi instalado um pequeno separador de fase. Esta é uma medida de segurança já que a válvula de expansão é manipulada manualmente.

CAPÍTULO III

Instrumentação e hardware de aquisição de dados

Este capítulo descreve a instrumentação utilizada na bancada experimental, incluindo uma breve descrição do hardware de filtragem de sinais, sensores utilizados e capítulo conclui com a especificação do sistema de aquisição e monitoramento de dados.

3.1 Hardware de filtragem de sinais

O sistema de medição e filtragem dos sinais dos sensores (pressão e temperatura) foi projetado e construído no Laboratório de Energia e Sistemas Térmicos (LEST) da Faculdade de Engenharia Mecânica (FEMEC) da UFU (Carvajal, Dissertação de Mestrado, 2000); Ele compõe-se basicamente de fontes de alimentação, comparadores e amplificadores operacionais, os quais serão descritos posteriormente com mais detalhes. Os circuitos analógicos construídos para o tratamento de sinais de temperatura e pressão são mostrados na Figura 3.1.

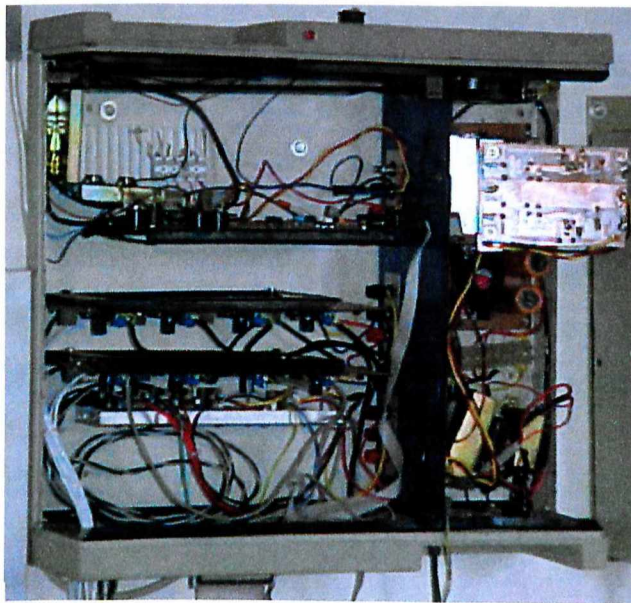


Figura 3.1 Circuito analógico.

3.2 Sistema de medição e filtragem

3.2.1 Fontes de alimentação

As fontes de alimentação fornecem tensões contínuas aos sensores e aos componentes do circuito de tratamento de sinais. O conjunto é formado por três fontes simétricas ajustáveis com voltagem de saída de 0 Vcc até ± 16 Vcc e projetadas para operar utilizando-se de reguladores de voltagem comerciais: LM 317 e LM 337.

3.2.2 Instrumentação e descrição dos sensores

O sistema de refrigeração, descrito no capítulo anterior, foi instrumentado com sensores de temperatura no circuito do fluido secundário e sensores de temperatura e pressão no circuito do fluido primário. Os sensores estão posicionados nas entradas e saídas dos principais componentes do sistema, como mostra a Figura 2.1, o que permite a análise do comportamento individual de cada componente.

3.2.2.1 Sensor de temperatura

Os sensores utilizados na bancada experimental foram termômetros de resistência (RTD – Resistance Temperature Detector) ou termoresistências PT 100, devido a sua grande sensibilidade, larga faixa de utilização e alta precisão. Seu funcionamento baseia-se na variação da resistência em função da temperatura. Os materiais mais utilizados para a fabricação destes tipos de sensores são platina, cobre ou níquel.

Entre esses materiais, o mais utilizado é a platina, pois apresenta uma ampla escala de temperatura, uma alta resistividade, permitindo assim uma maior sensibilidade, um alto coeficiente de variação de resistência com a temperatura, uma relação quase linear entre a resistência e a temperatura.

A variação da resistência em função da temperatura é regida pela equação 3.1, através do chamado coeficiente de temperatura da resistência (α) característico do material utilizado na sonda resistiva.

$$R_t = R_0(1 + \alpha T) \quad (3.1)$$

Onde:

R_t : Resistência a T °C [Ω]

R_0 : Resistência a 0 °C [Ω]

T : Temperatura [°C]

3.2.2.1.1 Princípio de medição

Para determinar o valor da resistência ôhmica das termoresistências, utiliza-se um circuito de medição tipo Ponte de Wheatstone. Devido a sua alta precisão é bastante utilizada em laboratórios e em algumas indústrias.

Existem três tipos de montagens da ponte: dois, três e quatro fios. Neste trabalho foi utilizada a ligação de três fios. Este tipo de montagem é a mais utilizada industrialmente. Neste circuito, a configuração elétrica é um pouco diferente, fazendo com que a alimentação fique o mais próximo possível do sensor, permitindo que a RL_1 passe para o outro braço da ponte, balanceando o circuito.

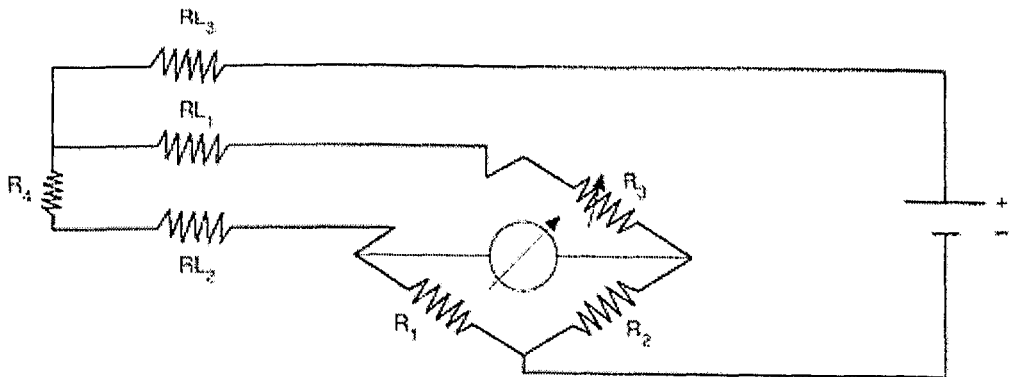


Figura 3.2 Ponte de Wheatstone Ligação três fios.

A Tabela 3.1 apresenta as especificações dos componentes do circuito da Figura 3.2.

Tabela 3.1 Especificações dos componentes da Ponte Wheatstone.

Componente	Valor
R1	200 Ω
R2	200 Ω
R3	250 Ω
V1	± 10 Vcc

Para o tratamento dos sinais dos 8 sensores PT-100 classe A, (bulbo cerâmico, 1 mm de diâmetro, 15 mm de comprimento), além da ponte de Wheatstone, utilizaram-se dois

amplificadores operacionais inversores ligados em série, que fornecem um ganho total de 1:46 e dois filtros passivos passa baixa de primeira ordem, ligados a cada um dos amplificadores (conforme a Figura 3.3).

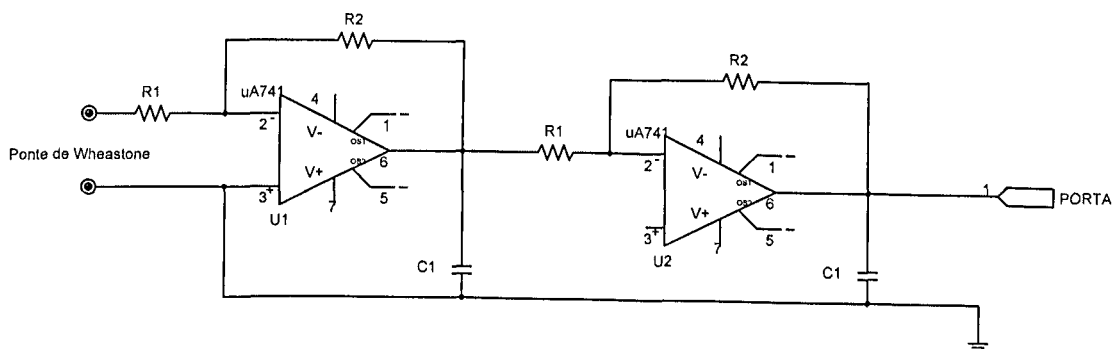


Figura 3.3 Circuito amplificador para o sensor PT-100.

A descrição dos elementos do circuito amplificador utilizado para os PT-100 encontra-se logo abaixo na Tabela 3.2.

Tabela 3.2 Elementos do circuito amplificador.

Componente	Valor
R1	1000 Ω
R2	6800 Ω
C1	22 μF
Circuito Integrado	LM 741

As tomadas de temperatura foram feitas em pontos estratégicos do sistema e em contra fluxo do refrigerante como mostra a Figura 3.4.

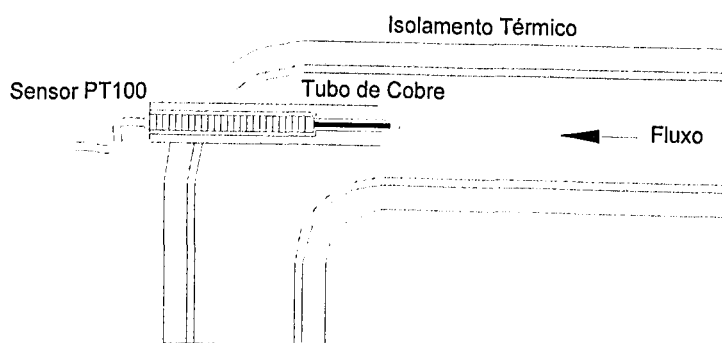


Figura 3.4. Tomadas de temperatura no interior do tubo.

3.2.2.2 Transdutor de pressão

Para a medição da pressão manométrica do fluido refrigerante nos pontos de entrada e saída de cada um dos componentes do sistema, utilizaram-se tradutores de pressão piezo-resistivos com faixa de operação de 0 - 10 e 0 - 25 [bar] para baixa e alta pressão respectivamente. Os sensores foram alimentados por uma fonte simétrica com tensão de $\pm 12V_{cc}$. A Figura 3.5 mostra a montagem realizada.

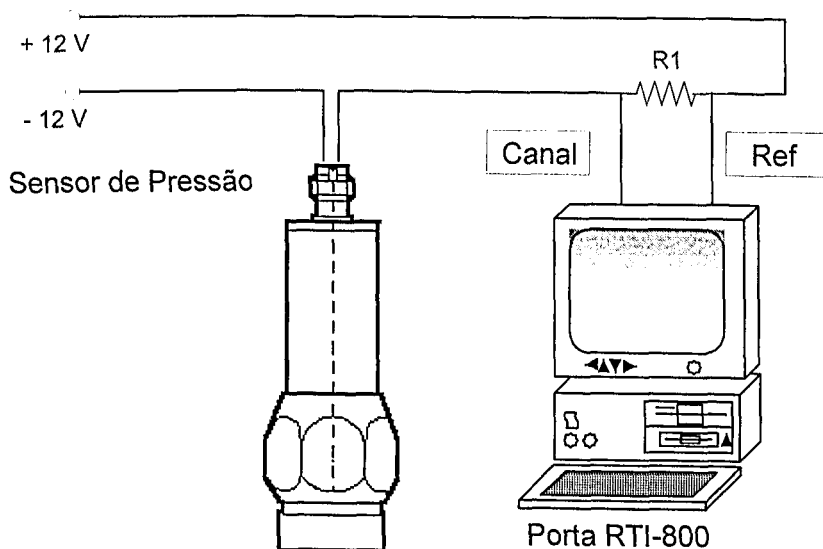


Figura 3.5 Esquema da montagem dos sensores de pressão.

Na Tabela 3.3 são apresentadas as especificações técnicas dos tradutores de pressão utilizados.

Tabela 3.3 Características técnicas dos sensores de pressão.

Voltagem de operação [Vcc]	10.8 até 30 DC
Faixa de corrente [mA]	4 até 20mA(600Ω)
Temperatura média [°C]	-25 até +80
Temperatura máxima [°C]	-40 até +100
Resolução [bar]	0.01

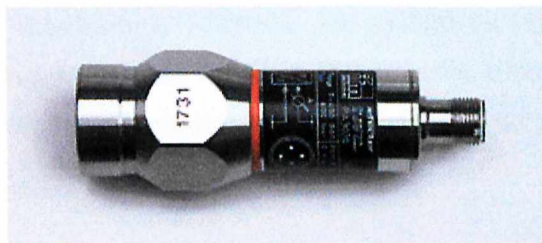


Figura 3.6 Foto do transdutor de pressão utilizado (Efactor PA3023).

3.3 Hardware de aquisição de dados

Para a comunicação da placa de tratamento de sinais com o microcomputador pessoal (PC) Pentium – 200 MHz, empregou-se uma placa de "Input/Output" RTI-800 da "Analog Divice" com configurações de pseudo-diferencial com faixas de voltagem de ± 10 Vcc. Foram utilizados 12 canais de entrada analógica. As características técnicas da placa são apresentadas em User's Manual RTI-800/815, 1988.

A conversão A/D (leitura dos sinais dos sensores de temperatura e pressão) obedece os seguintes requisitos:

- Configuração do ganho e seleção do sensor a ser lido
- Inicialização da conversão A/D
- Processamento de conversão A/D com um certo intervalo de tempo e
- Leitura do valor na porta A/D.

A configuração do ganho do amplificador da placa RTI-800 foi programada para um valor de 1 numa faixa de ± 10 volts através do endereço (base+1)

3.4 Software de monitoração

O software de monitoração foi implementado em C++ Builder da Borland para a aquisição de dados e controle. Ele consiste num sistema de monitoramento "on line" do circuito de refrigeração e apresenta uma interface gráfica que facilita ao usuário analisar diretamente o processo.

Os sinais adquiridos dos sensores de temperatura e pressão (volts) são convertidos nas unidades de medida correspondentes através das respectivas curvas de calibração, implementadas dentro do programa de aquisição de dados. Estes serão apresentados posteriormente. O arquivo de armazenamento destes dados é atualizado a cada tempo de aquisição de 1 [seg.].

As figuras a seguir mostram a interface do programa, que mostram o ciclo de compressão a vapor. Na entrada e saída de cada componente é mostrada a temperatura em cor preta e, em azul, as pressões absolutas. A figura do ciclo é comum em todas as telas do programa.

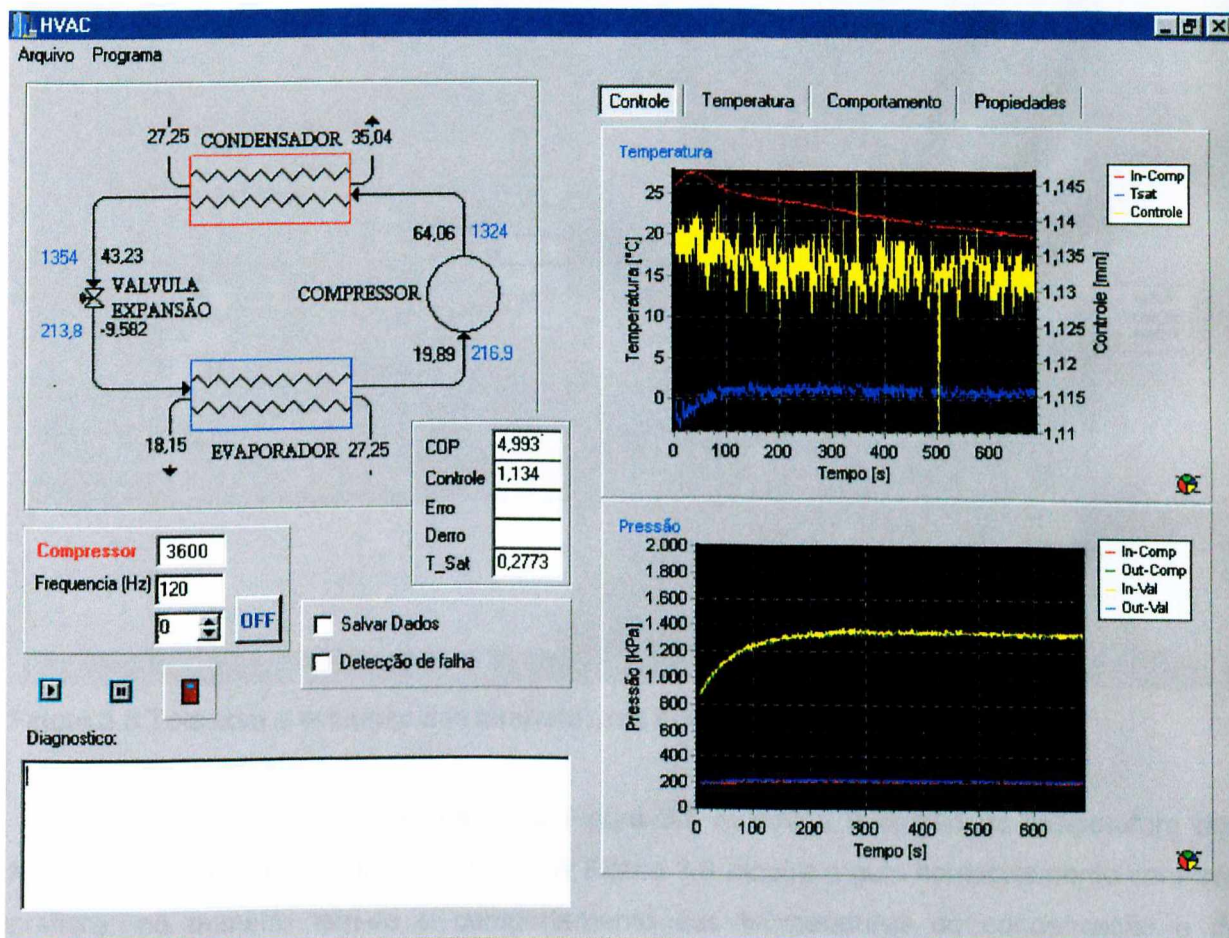


Figura 3.7 Tela principal do programa.

Do lado direito da interface temos a opção de visualizar 4 guias. A tela com a guia "controle" é mostrada na Figura 3.7. Nela pode-se ver o gráfico das principais variáveis relacionadas ao controle fuzzy e a evolução das pressões na entrada e saída de cada componente.

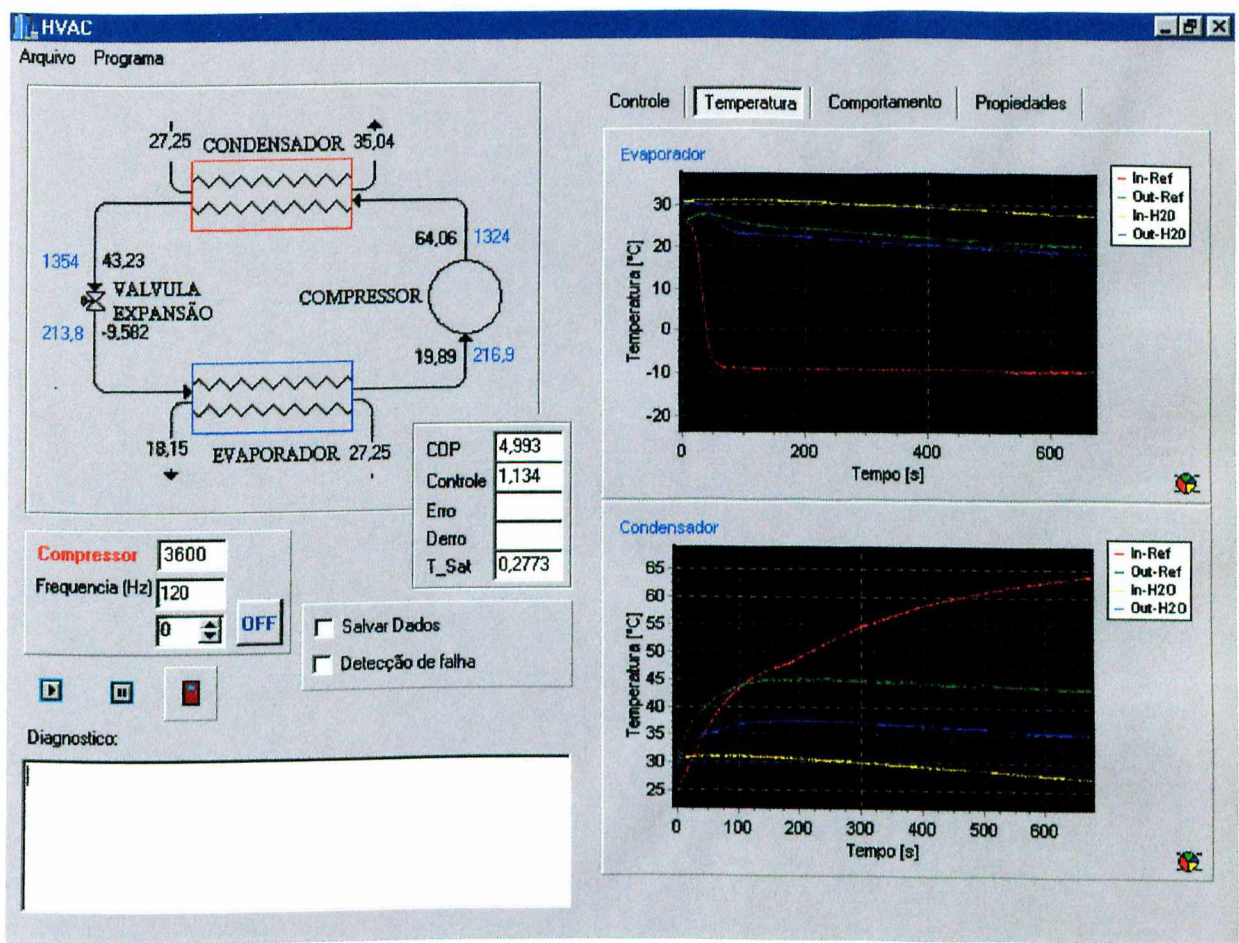


Figura 3.8 Tela com a evolução das temperaturas nos trocadores.

A guia Temperatura mostrada na Figura 3.8 mostra a evolução da temperatura dos fluidos no condensador e no evaporador. A Figura 3.9 mostra a guia comportamento com três gráficos: no primeiro tem-se o comportamento das temperaturas de condensação e de evaporação do ciclo, o segundo apresenta a diferença de temperatura entre a entrada e saída da água nos trocadores, no último gráfico mostra-se os parâmetros *APPR_e* (diferença de temperatura evaporação e temperatura da saída da água no evaporador) e *APPR_c* (diferença de temperatura condensação e temperatura da saída da água no condensador).

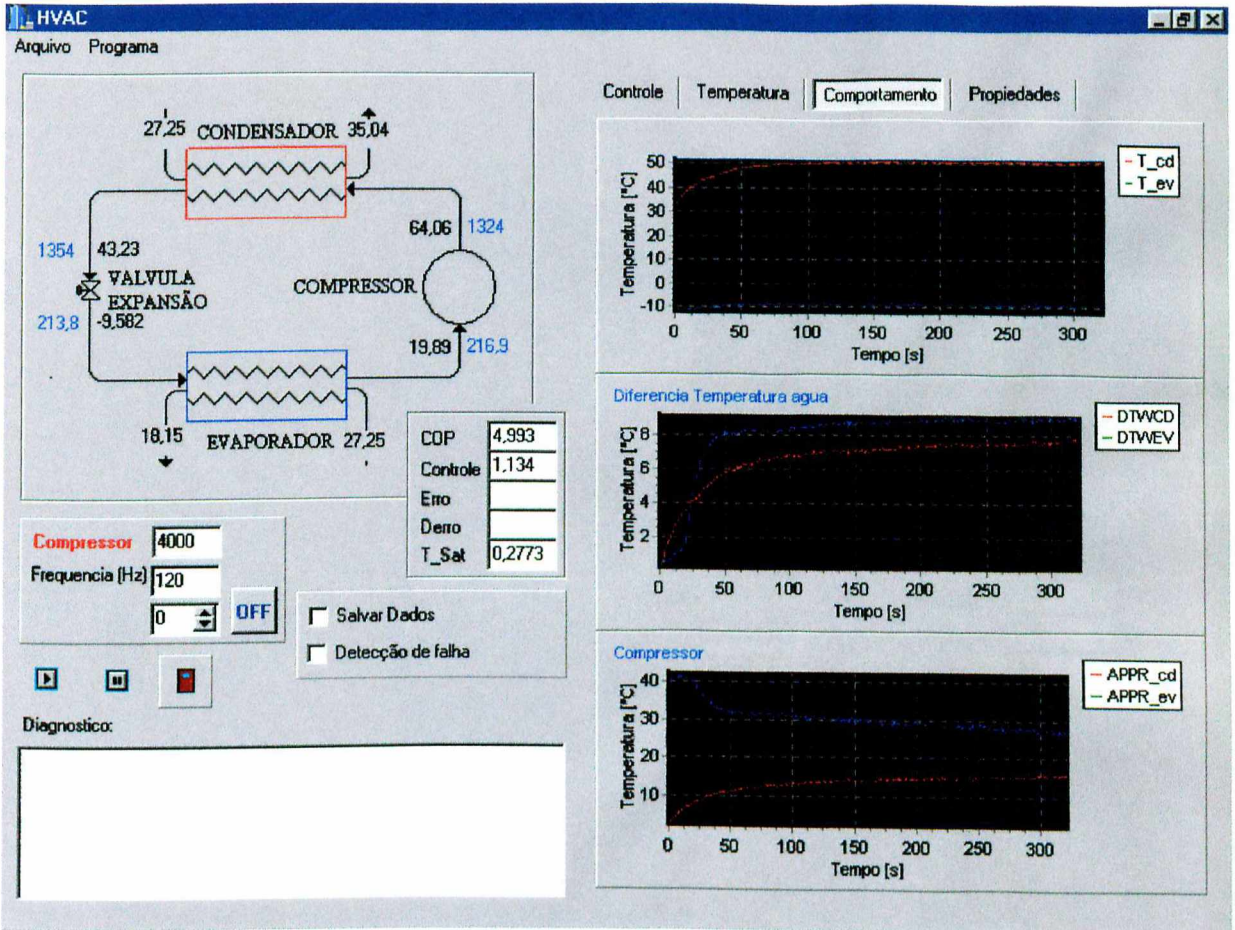


Figura 3.9 Tela com a evolução de alguns parâmetros importantes na ANN.

Na Figura 3.10, mostra-se o cálculo do pré-processamento térmico. O cálculo é feito realizando uma média de 10 em 10 pontos, para atenuar o efeito de oscilações nos sensores. O usuário deve fornecer a vazão de água nos trocadores de calor, para que se possa calcular vazão do refrigerante. As propriedades do fluido refrigerante são calculadas em tempo real pelo programa. Na figura observa-se também o diagrama P x h do ciclo.

Para o cálculo das propriedades do fluido utiliza-se uma rotina específica de cálculo de propriedades termodinâmicas (Type 58) presente no TRNSYS 14.1, implementada em Fortran 77. Foi necessário criar uma biblioteca dinâmica para que possa ser utilizada a rotina do TRNSYS no C++ Builder

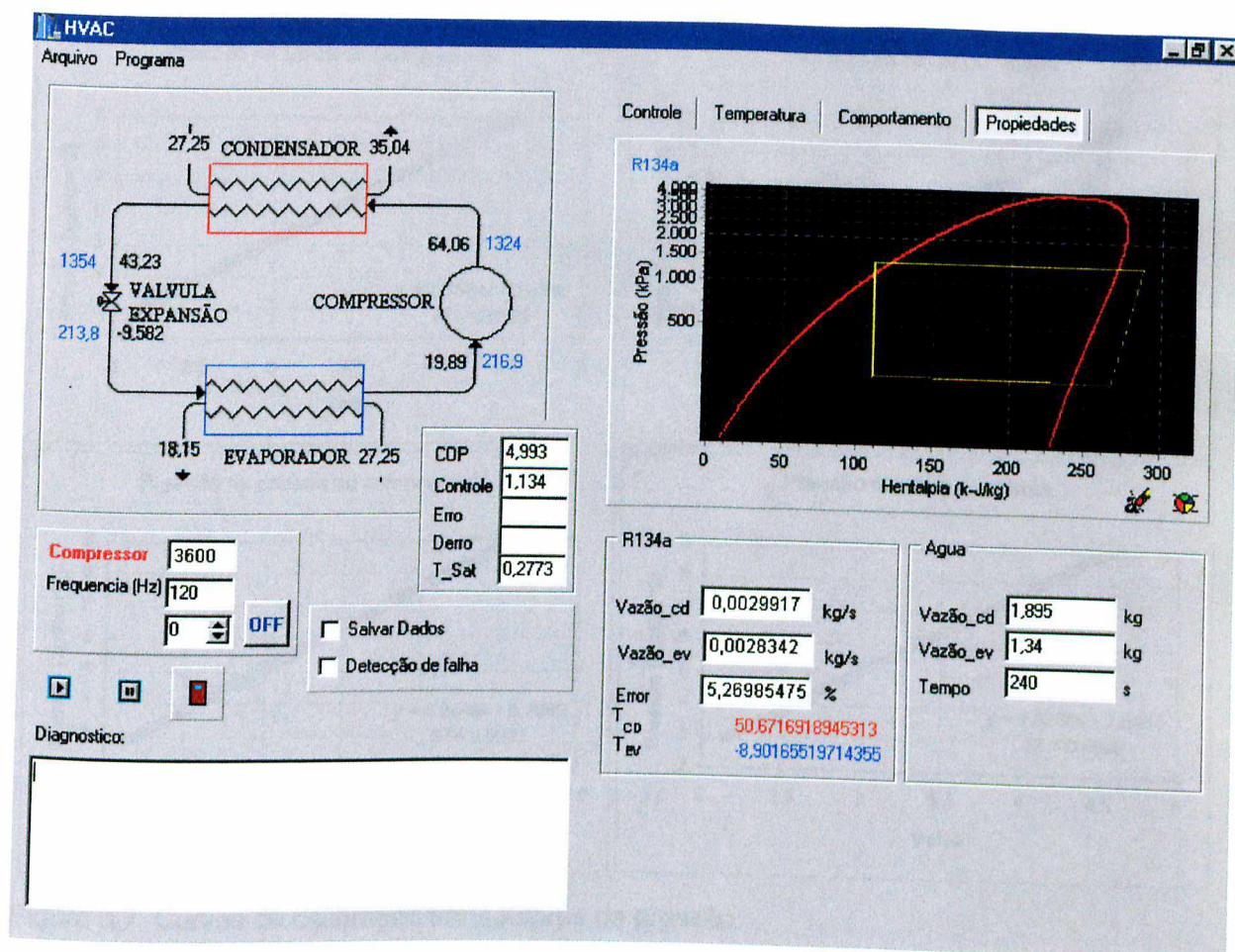


Figura 3.10 Tela com o cálculo das propriedades térmicas.

3.4.4 Curvas de calibração

Os sinais dos sensores (volts) são convertidos para as grandezas físicas medidas (temperatura, pressão) através das curvas de calibração das mesmas.

3.4.4.1 Curvas de calibração dos Transdutores de pressão

Para a calibração dos sensores de pressão, utilizou-se um manômetro padrão previamente calibrado pelo método da balança de calibração, cedida pela companhia Souza Cruz de Uberlândia com escala de $0,05 \text{ [kg/cm}^2\text{]}$ para uma faixa de 0 a 10 $[\text{kg/cm}^2]$. As curvas e equações de calibração para os quatro sensores de pressão instalados na entrada e saída da válvula e do compressor são apresentadas na Figura 3.7.

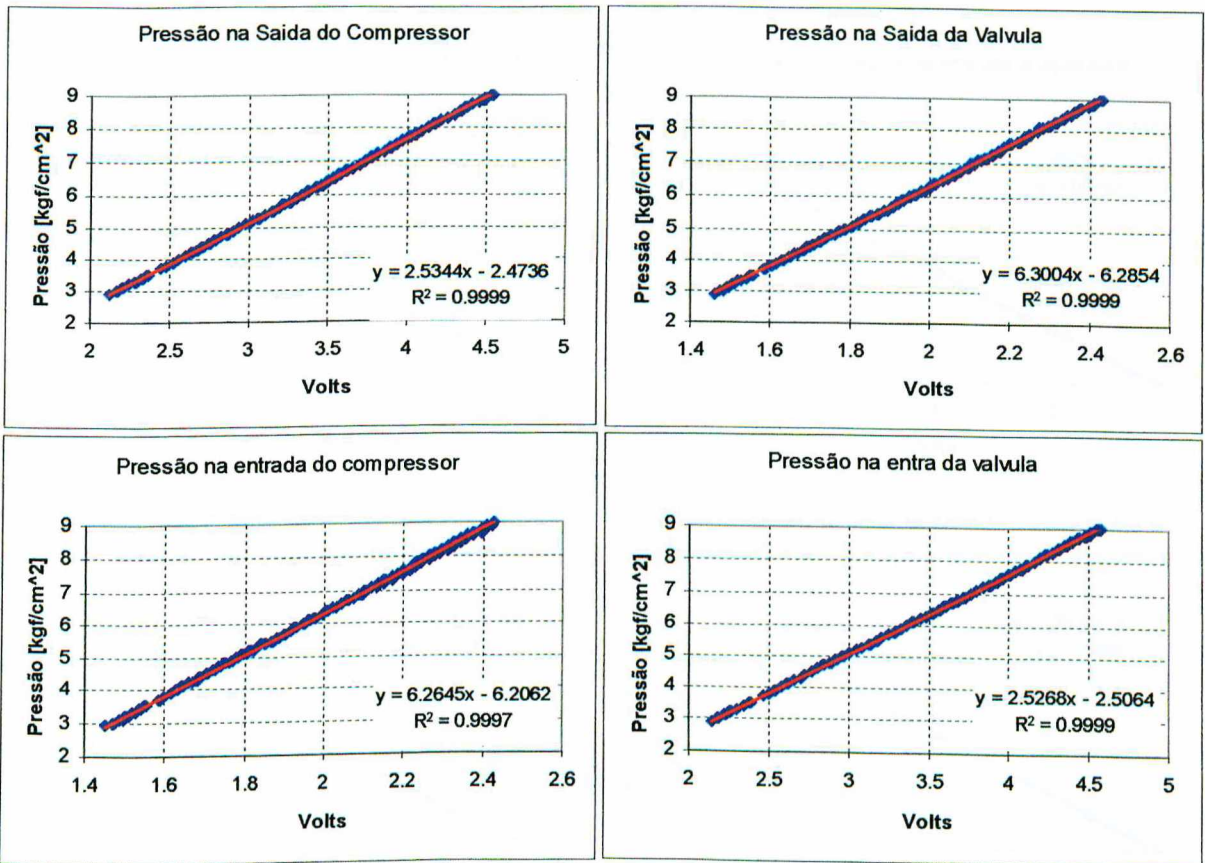


Figura 3.7. Curvas de calibração transdutores de pressão.

3.4.4.2 Curvas de calibração dos Sensores de Temperatura

Para calibrar os sensores de temperatura (PT-100), utilizou-se um termômetro padrão de mercúrio com precisão de 0.1 [°C]. Para cada sensor de temperatura tomaram-se os sinais (volts) e seus respectivos valores de temperaturas na faixa de 8 a 55 [°C]. Uma regressão linear foi realizada para determinar a equação de calibração correspondente a cada sensor de temperatura, apresentadas nas Figuras 3.8a e 3.8b.

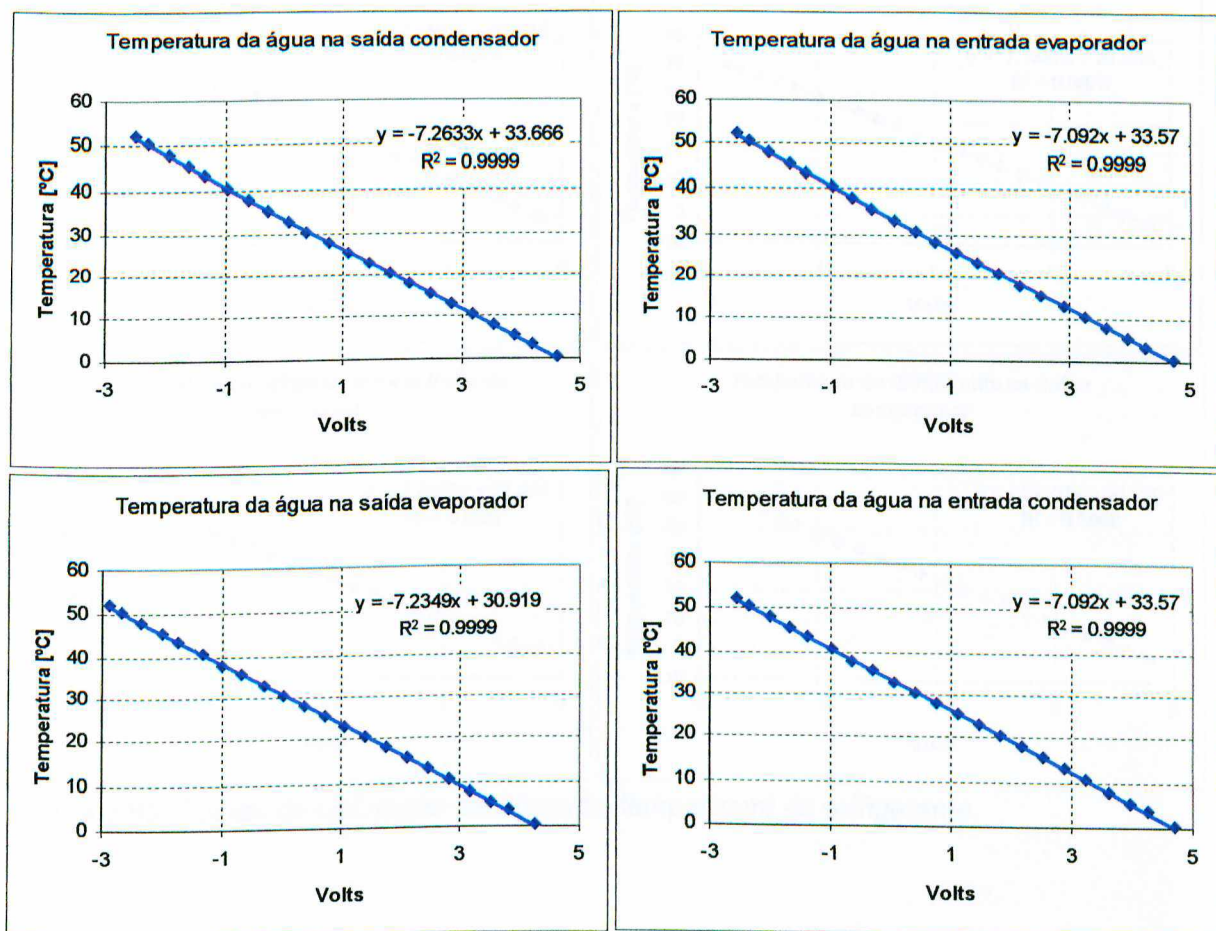


Figura 3.8a Curvas de calibração sensores de temperatura da água.

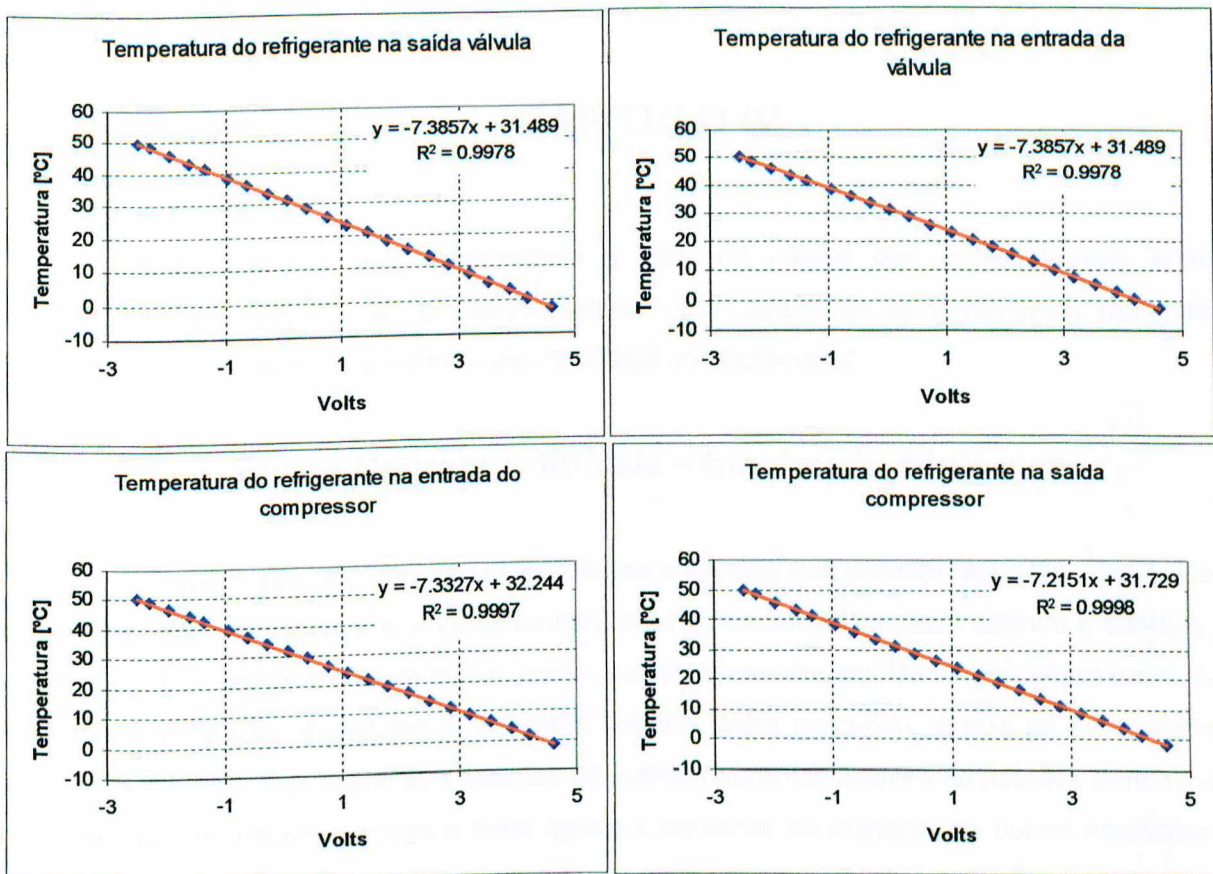


Figura 3.8b. Curvas de calibração sensores de temperatura do refrigerante.

CAPÍTULO IV

Neste capítulo será apresentado o conceito básico das redes neurais artificiais utilizadas. No início do capítulo descrevem-se detalhadamente as ferramentas matemáticas básicas usadas para a implementação do código computacional.

Redes Neurais Artificiais – Inspiração Biológica

Ninguém sabe exatamente de que forma o cérebro nos permite raciocinar. Sabemos, no entanto, que o neurônio é a unidade fundamental do Sistema Nervoso incluindo o cérebro. Um neurônio é constituído por um corpo celular que contém o núcleo. Do corpo celular saem vários “ramos” chamados **dendritos** e um ramo isolado, mais comprido, a que se dá o nome de **axônio**. Os dendritos ligam-se a axônios de outros neurônios através de junções denominadas **sinápses**. Um neurônio chega a estar ligado a centenas de milhares de outros neurônios. Os sinais propagam-se entre os neurônios através de uma reação eletroquímica complicada que faz com que as sinápses libertem substâncias químicas (neuro transmissores) que entram pelos dendritos fazendo subir ou descer o potencial elétrico do corpo celular. Quando este potencial ultrapassa um determinado limite é enviado um impulso elétrico pelo axônio que se “espalha” pelas ramificações deste, transmitindo sinais elétricos a outros neurônios. Já se observou que as ligações mais usadas se tornam mais fortes e que os neurônios por vezes formam novas ligações com outros neurônios, portanto julga-se que esses mecanismos permitem a aprendizagem.

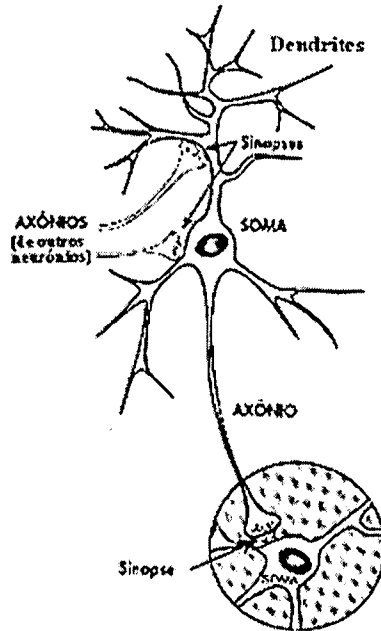


Figura 4.1 Neurônio biológico.

4.1 Neurônio artificial

O neurônio artificial é uma estrutura lógico-matemática que procura simular a forma, o comportamento e as funções de um neurônio biológico, que possui múltiplas entradas e uma única saída.

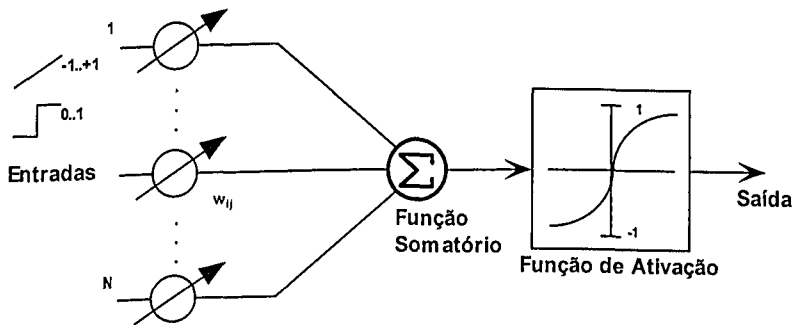


Figura 4.2 - Neurônio artificial simplificado

O neurônio artificial avalia seus sinais de entrada realizando uma somatório ponderado das suas entradas (através dos pesos "sinápticos" associado a cada entrada) como mostra a Equação 4.1:

$$I_j = \sum_{i=0}^N w_{ij} x_i \quad (4.1)$$

Onde I_j representa a soma ponderada dos N sinais de entrada do neurônio j , w_{ij} representa o valor do peso sináptico associado a cada conexão entre o neurônio i e j , e x_i representa a saída do i -ésimo neurônio.

De uma maneira mais simplificada, isto significa somar todos os sinais de entrada que chegam a um neurônio, levando em consideração o peso das conexões envolvidas em cada sinal de entrada. O sinal de saída do neurônio artificial é encontrado aplicando-se o somatório ponderado das suas entrada numa função de ativação que determina seu valor de saída (nível de ativação)

$$Y_j = \phi(I_j) \quad (4.2)$$

Onde Y_j é a saída do neurônio, ϕ é a função de ativação do neurônio.

A Figura 3 mostra algumas formas de função de ativação ou função de transferência de um neurônio.

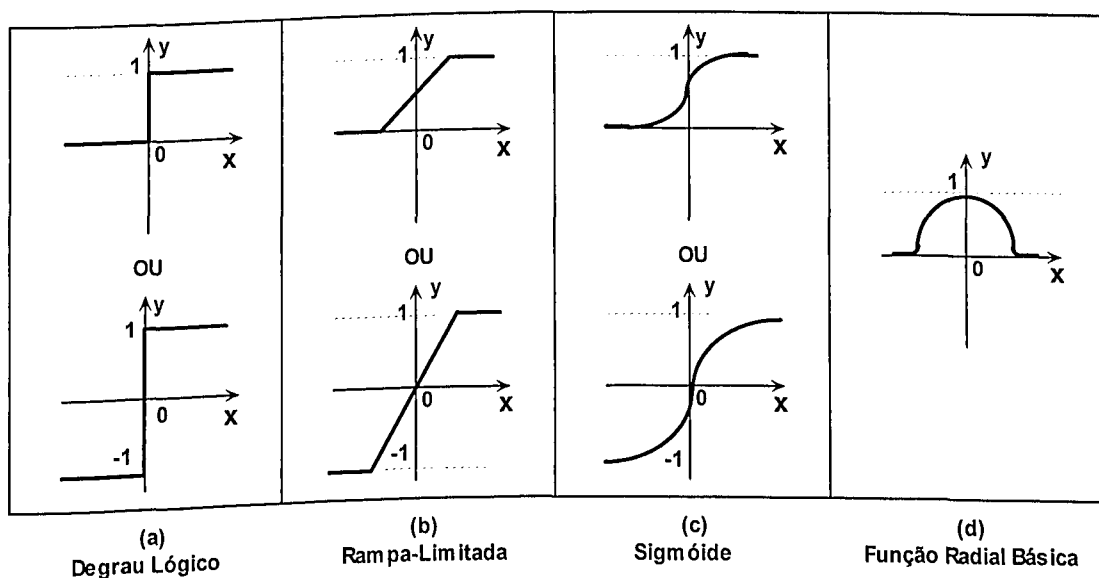


Figura 4.3. Típicas funções de transferência utilizadas nas redes neurais.

As funções ativação mais utilizadas são as funções sigmóides, como arco-tangente, a tangente hiperbólica etc, que variam gradualmente entre as assíntotas: $\phi(I) = 0$ e $\phi(I) = 1$ ou $\phi(I) = -1$ e $\phi(I) = 1$.

Entretanto, podem ser utilizadas funções de ativação do tipo linear: $\phi(I) = I$. Salienta-se que o uso de uma função de ativação linear remove a não linearidade do neurônio artificial, impossibilitando as redes neurais à identificar os fenômenos não lineares.

4.2 Redes neurais artificiais

Uma rede neural completa é organizada na forma de camadas. Uma rede pode possuir n neurônios na camada de entrada, m neurônios na camada seguinte, e assim sucessivamente até a camada final de saída. Uma rede com mais de uma camada pode ser caracterizada como uma **rede multicamadas**.

A forma pela qual os neurônios estão conectados uns aos outros (topologia ou arquitetura da rede) causa um enorme efeito na operação da rede neural. As camadas de uma rede neural são interconectadas através de parâmetros internos denominados pesos (w). A camada de entrada somente apresenta os dados à rede neural. Ela não possui neurônio de processamento. A camada de saída resulta nos valores de saída de rede. As outras camadas são chamadas de intermediárias ou ocultas. A arquitetura da rede neural mostrada na Figura 4.4, por exemplo, é composta de uma camada de entrada com três neurônios, uma camada oculta com cinco neurônios e uma camada de saída com dois neurônios.

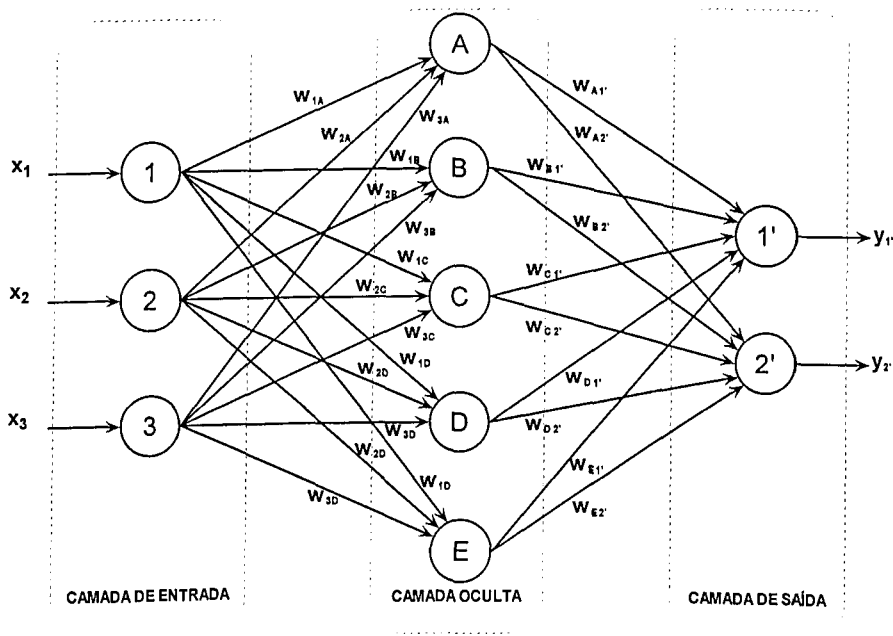


Figura 4.4 arquitetura de uma rede neural *feedforward* multicamadas.

4.3 Treinamento das redes neurais artificiais

O treinamento da rede neural é feito através de um processo iterativo de ajustes aplicado a seus pesos, até que a rede neural atinja a saída desejada para uma determinada entrada. Existem dois tipos distintos de treinamento:

- (i) **Treinamento não supervisionado ou auto-organizado.** Neste treinamento a rede não recebe nenhuma informação de como classificar as entradas e ajustar os pesos.
- (ii) **Treinamento supervisionado ou associativo.** As redes comumente utilizadas em problemas de engenharia envolvem treinamento supervisionado. Este treinamento recebe este nome pois a rede artificial depende de um sistema auxiliar para supervisionar o treinamento, ou seja é necessário um conjunto de dados experimentais confiáveis, ambos utilizados na avaliação da capacidade da rede em mapear as entradas e saídas. No treinamento supervisionado, o vetor das variáveis de entrada possui um correspondente vetor de variáveis de saída.

4.4 Redes backpropagation

A rede neural baseada no algoritmo de treinamento *backpropagation* é a mais difundida (Hammerstron, 1993; Touretzky, 1989). Trata-se de uma rede perceptron (não-linear) de multicamadas *feedforward*, onde nenhuma informação é retroprogramada durante sua operação (Antsaklis, 1992). É assim denominada pelo seu esquema de treinamento

supervisionado, no qual um sinal de erro de saída é retropropagado pela rede modificando o peso das conexões de forma a minimizar este erro.

Uma rede *backpropagation* requer no mínimo três camadas que são usualmente referenciadas como camada de entrada, oculta ou intermediária e de saída.

A Figura 4.5. ilustra uma rede multicamadas com arquitetura do tipo *feedforward*, com as camadas mencionadas anteriormente. Normalmente adiciona-se às camadas de entrada e às ocultas um neurônio extra chamado de *bias* de valor unitário. Todos os neurônios são interconectados através dos pesos.

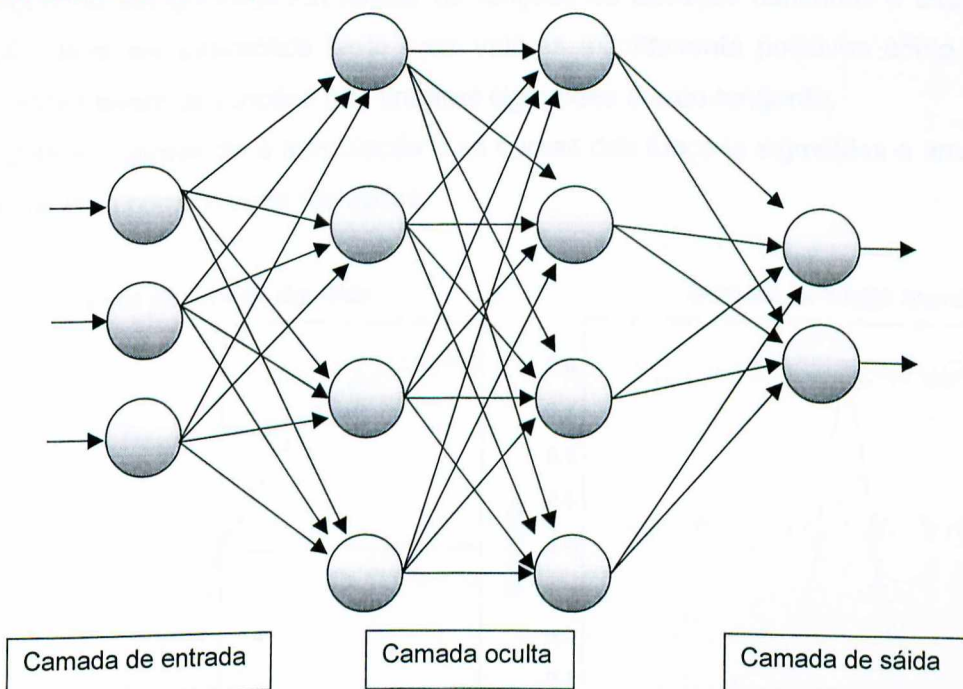


Figura 4.5. Arquitetura de uma rede neural artificial com quatro camadas.

O modelo típico do neurônio utilizado no algoritmo de *backpropagation* é apresentado na Figura 4.6, onde as entradas x_n são conectadas pelos seus respectivos pesos w_n para o processamento de sua saída por intermédio da função de ativação $\phi(I)$.

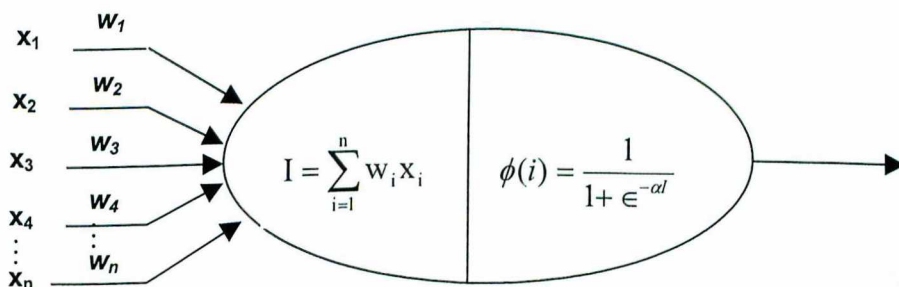


Figura 4.6. Neurônio de processamento do algoritmo *backpropagation* com função de ativação logística.

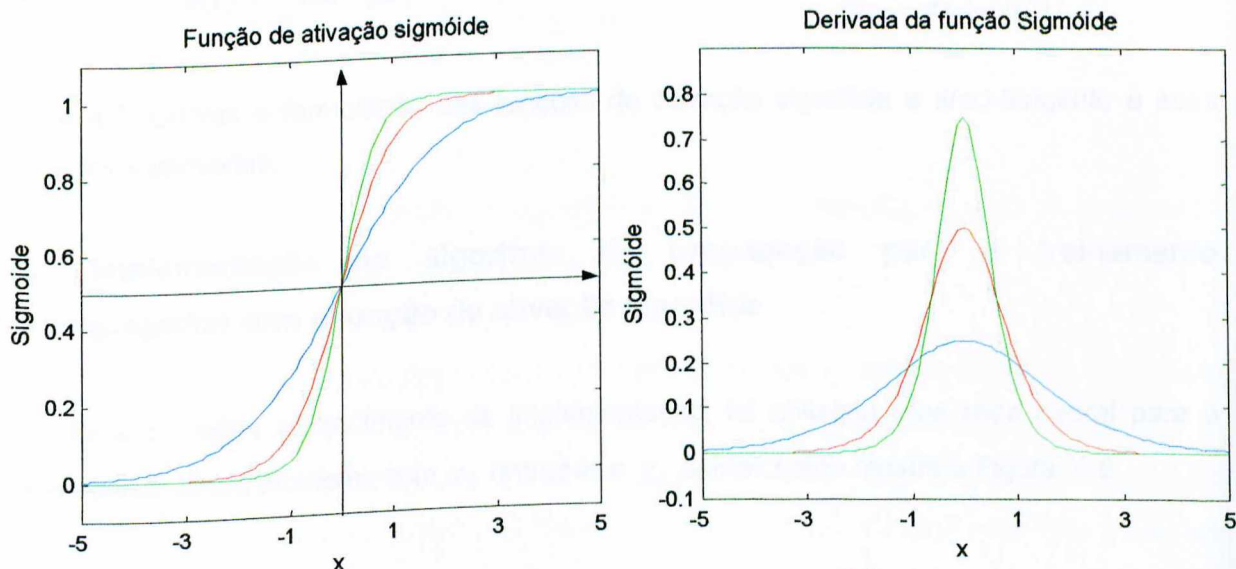
4.41 Função de ativação para o algoritmo *backpropagation*

As entradas de uma função de ativação são as somatórias dos produtos dos pesos pelas respectivas entradas.

$$w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n = \sum_{i=1}^n w_ix_i = I \quad (4.3)$$

O algoritmo *backpropagation* requer de funções de ativação contínuas e diferenciáveis. Esta função deve ser assintótica tanto para valores infinitamente positivos como negativos. Estas condições levam às funções não lineares sigmóides e arco-tangente.

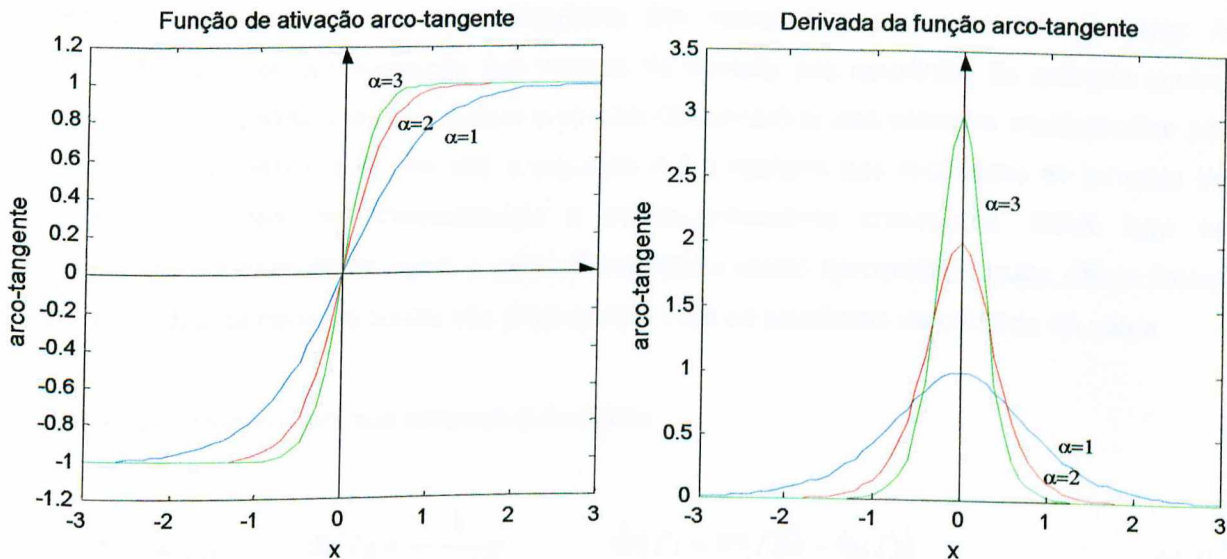
A Figura 4.7 apresenta a formulação e as curvas das funções sigmóides e arco-tangente mencionadas, bem como a suas derivadas.



(a)

$$\phi(I) = \frac{1}{1 + e^{-aI}}$$

$$\frac{\partial \phi_{jm}}{\partial I} = \phi[1 - \phi_{jm}]$$



(b)
$$\phi(I) = \frac{2}{\pi} \tan^{-1}(\alpha I)$$

$$\frac{\partial \phi(I)}{\partial I} = \frac{2}{\pi} \left[\frac{\alpha}{1 - \alpha^2 I^2} \right]$$

Figura 4.7. Curvas e formulação das funções de ativação sigmóide e arco-tangente e suas respectivas derivadas.

4.4.2 Implementação do algoritmo de propagação para o treinamento *backpropagation* com a função de ativação sigmóide

Para o melhor entendimento da implementação, foi utilizada uma rede neural para o mapeamento de um problema com x_n entradas e y_n saídas como mostra a Figura 4.8.

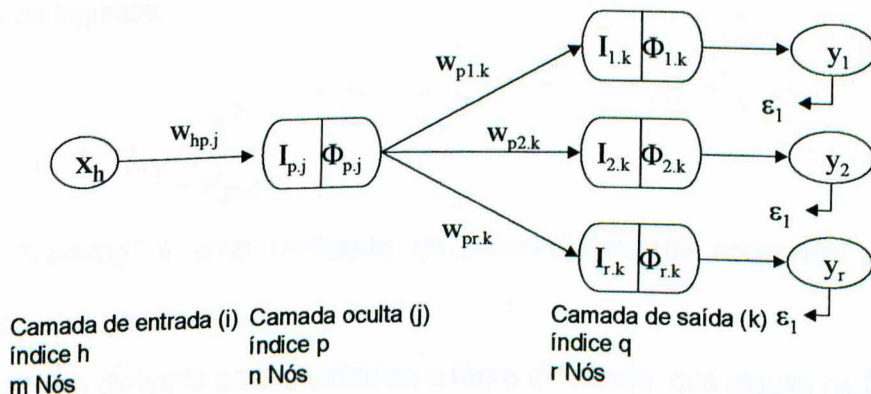


Figura 4.8 Esquema de rede neural com uma camada oculta, mostrando os símbolos e os índices.

1° Passo: Inicia-se com a geração aleatória dos pesos com pequenos valores reais. A propagação parte da apresentação dos valores de entrada aos neurônios da camada oculta. Os neurônios da camada oculta iniciam o cálculo da somatória das entradas multiplicadas por seus respectivos pesos como mostra a equação 4.3 e aplicam aos resultados às funções de ativação que devem ser diferenciáveis e monotonicamente crescentes. Estas logo se propagam para a camada de saída a partir do resultado obtido na camada oculta. Desta forma todas as saídas da camada oculta são propagadas para os neurônios da camada de saída.

A função de ativação com sua respectiva derivada.

$$\text{Sigmóide} \quad \Phi(I) = \frac{1}{1 + e^{-aI}} \quad \dot{\Phi}(I) = \Phi(I)(1 - \Phi(I)) \quad (4.4)$$

Uma vez propagadas as entradas da rede, define-se o erro médio quadrático:

$$\varepsilon^2 = [Y_r - \Phi_{rk}]^2 \quad (4.5)$$

Onde Y_r são as saídas desejadas e Φ_{rk} são as saídas propagadas da rede, A atualização dos pesos se realizam no segundo passo do treinamento .

2° Passo: Refere-se à atualização de pesos na camada de saída. São mudadas em proporção ao erro quadrático médio em relação aos respectivos pesos. As equações seguintes são as derivações para uma rede de duas camadas (oculta e saída) com a função de ativação logística. Note que as saídas podem, somente, ter uma escala [0...1] para uma rede com uma camada saída logística.

$$\Delta w_{pq,k} = -\eta_{p,q} \frac{\partial \varepsilon^2}{\partial w_{pq,k}} \quad (4.6)$$

onde $\eta_{p,q}$ "learning" é uma constante de proporcionalidade associado a uma taxa de aprendizagem dos pesos $w_{pq,k}$.

Para resolver esta derivada parcial utiliza-se a regra da cadeia que resulta na Equação (4.7).

$$\Delta w_{pq,k} = -\eta_{p,q} \frac{\partial \varepsilon^2}{\partial \Phi_{q,k}} \frac{\partial \Phi_{q,k}}{\partial I_{q,k}} \frac{\partial I_{q,k}}{\partial w_{pq,k}} \quad (4.7)$$

Onde: A derivada do erro em relação a função de ativação é dada por:

$$\frac{\partial \varepsilon^2}{\partial \Phi_{q,k}} = (-2)[T_q - \Phi_{q,k}] \quad (4.7a)$$

A derivada da função de ativação em relação a somatórias dos produtos dos pesos com suas respectivas entradas é dada por:

$$\frac{\partial \Phi_{q,k}}{\partial I_{q,k}} = \Phi_{q,k} \quad (4.7b)$$

A derivada das somatórias dos produtos dos pesos com suas respectivas entradas em relação aos pesos é dada por:

$$\frac{\partial I_{q,k}}{\partial w_{pq,k}} = [1 - \Phi_{q,k}] \cdot \Phi_{p,j} \quad (4.7c)$$

A substituição dessa derivadas na equação 4.7 resulta em:

$$\Delta w_{pq,k} - \eta_{p,q} \cdot (-2)[T_q - \Phi_{q,k}] \cdot \Phi_{q,k} [1 - \Phi_{q,k}] \cdot \Phi_{p,j} = \eta_{p,q} \cdot \delta_{pq,k} \cdot \Phi_{p,j} \quad (4.7d)$$

Onde:

$$\delta_{pq,k} = 2[T_q - \Phi_{q,k}] \Phi_{q,k} [1 - \Phi_{q,k}] \quad (4.7e)$$

Logo os pesos são atualizados conforme a equação 4.8, de recorrência:

$$w_{pq,k}(N+1) = w_{pq,k}(N) + \eta_{p,q} \cdot \delta_{pq,k} \cdot \Phi_{p,j} \quad (4.8)$$

Note que N é número de iterações ou épocas.

Para o cálculo dos pesos que estão entre a camada de entrada e a camada oculta utilizam-se as seguintes derivadas. (relembro que as três primeiras derivadas parciais resultantes já foram calculadas como mostra a equação 4.8, falta o calculo das duas ultimas derivadas).

$$\Delta w_{hp,j} = -\eta_{h,p} \frac{\partial \varepsilon^2}{\partial w_{hp,j}} \quad (4.9)$$

Onde $\eta_{h,p}$ é outra constante proporcional de aprendizagem relativa à atualização dos pesos $w_{hp,j}$ novamente utilizando a regra da cadeia a equação 4.9, pode ser escrita;

$$\Delta w_{hp,j} = -\eta_{h,p} \frac{\partial \varepsilon_q^2}{\partial \Phi_{q,k}} \cdot \frac{\partial \Phi_{q,k}}{\partial I_{q,k}} \cdot \frac{\partial I_{q,k}}{\partial \Phi_{p,j}} \cdot \frac{\partial \Phi_{p,j}}{\partial I_{p,j}} \cdot \frac{\partial I_{p,j}}{\partial w_{hp,j}} \quad (4.10)$$

Onde: As duas derivadas restantes são dadas por:

$$\frac{\partial \Phi_{p,j}}{\partial I_{p,j}} = \Phi_{p,j} [1 - \Phi_{p,j}] \quad (4.11)$$

$$\frac{\partial I_{p,j}}{\partial w_{hp,j}} = x_h \quad (4.12)$$

A substituição dessas derivadas na equação 4.10 resulta em:

$$\frac{\partial \varepsilon^2}{\partial w_{hp,j}} = (-2) [I_q - \Phi_{q,k}] \cdot \Phi_{q,k} [1 - \Phi_{q,k}] \cdot w_{pq,k} \cdot \Phi_{p,j} [1 - \Phi_{p,j}] x_h \quad (4.13)$$

ou:

$$\frac{\partial \varepsilon^2}{\partial w_{hp,j}} = \delta_{pq,k} \cdot w_{pq,k} \cdot \Phi_{p,j} [1 - \Phi_{p,j}] x_h \quad (4.13a)$$

onde:

$$\delta_{hp,j} = \delta_{pq,k} w_{pq,k} \frac{\partial \Phi_{p,j}}{\partial I_{p,j}} \quad (4.13b)$$

logo os pesos $w_{hp,j}$ são atualizados conforme a equação 4.14, de recorrência:

$$w_{hp,j}(N+1) = w_{hp,j}(N) + \eta_{hp} x_h \delta_{hp,j} \quad (4.14)$$

Onde N é o número de iterações.

Para cada um dos pesos das interconexões entre as camadas de entrada e a camada oculta ou entre a camada oculta e a camada de saída, é aplicado o processo, descrito nas sessões anteriores, de ajuste de pesos.

Nas duas equações 4.8 e 4.14 aparece o termo η referente à taxa de aprendizado. Esta variável determina pequenas mudanças nos pesos da rede de uma iteração para a outra. Esta variável, inicialmente estimada também é atualizada de iteração a iteração seguindo o critério definido por Wiley & Sons(1994). Para o incremento e decremento do η é feito o seguinte teste durante o transcurso das iterações:

Se erro calculado está próximo a erro desejado
Então taxa de aprendizado é reduzida
Se não taxa de aprendizado é incrementada

A idéia deste teste é manter a mudança de seus pesos proporcionais a suas variações. Na prática, utiliza-se uma taxa de aprendizado de elevada magnitude que proporciona poucas oscilações na direção de busca.

Uma outra maneira de evitar bruscas oscilações, sem a necessidade de escolher uma taxa de aprendizado pequena, é a inserção do termo *momentum*, assim as equações ficam rescritas:

$$w_{pq,k}(N+1) = \alpha \cdot w_{pq,k}(N) + \eta_{pq} \cdot \delta_{pq,k} \cdot \Phi_{p,j} \quad (4.15)$$

$$w_{hp,j}(N+1) = \alpha \cdot w_{hp,j}(N) + \eta_{hp} x_h \delta_{hp,j}$$

(4.16)

onde α é usualmente um número positivo chamado constante de *momentum*, e controla o ciclo de realimentação da variação do peso.

4.42 Metodologia utilizada para o treinamento da rede neural

Para a realização desta etapa, foram necessários os seguintes passos que são apresentados no fluxograma a seguir:

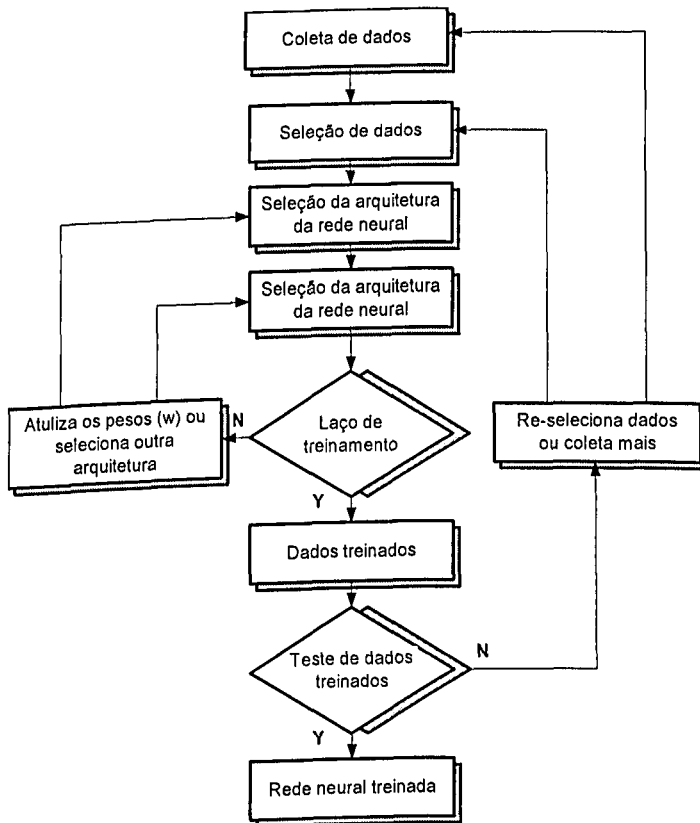


Figura 4.9 Fluxograma da metodologia do algoritmo de treinamento.

O código computacional foi implementado em linguagem C++ com o objetivo de ser utilizado no programa de monitoramento. As rotinas do programa serão apresentadas em anexo.

CAPÍTULO V

Neste capítulo será apresentado o ciclo de refrigeração por compressão a vapor, em seguida o modelo de detecção de falha, identificação e análise de sensibilidade dos parâmetros para uma falha específica, resultados do treinamento da ANN para funcionamento normal e com falha e finalmente a detecção e diagnóstico das falhas usando esta metodologia.

5.1 Ciclo padrão de compressão a vapor

O ciclo padrão de refrigeração por compressão de vapor é mostrado na Figura 5.1, construído sobre um diagrama de Mollier no plano P-h.

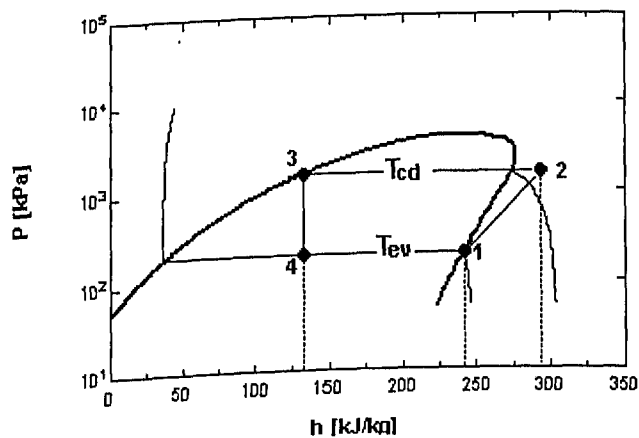


Figura 5.1 Ciclo padrão de refrigeração por compressão a vapor

Os processos termodinâmicos que constituem o ciclo teórico, e seus respectivos equipamentos são:

- Processos 1→2, que ocorre no compressor, é o processo de compressão adiabático reversível. Neste processo a entropia (S) permanece constante, ou seja, $S_1=S_2$, como mostra a Figura 5.1. O refrigerante entra no compressor à pressão do evaporador (P_{ev}) e com título $X=1$. O refrigerante é então comprimido até atingir a pressão de condensação. O refrigerante, na saída do compressor, encontra-se superaquecido com temperatura T_2 , superior à temperatura de condensação (T_{cd}).

- b) Processo 2→3, ocorre no condensador. É um processo de rejeição de calor do refrigerante para o meio de resfriamento, desde a temperatura T_2 de saída do compressor até atingir o estado de líquido saturado na temperatura de saturação (T_{cd}), correspondente a pressão de condensação (P_{cd}).
- c) Processo 3→4, ocorre na válvula de expansão. É uma expansão irreversível com entalpia constante desde a P_{cd} e líquido saturado ($X=0$) até atingir a pressão do evaporador P_{ev} . O processo é irreversível, e portanto, a entropia (S) do refrigerante ao deixar a válvula de expansão (S_4) é maior que a entropia do refrigerante ao entrar na válvula (S_3).
- d) Processo 4→1, ocorre no evaporador. É um processo de transferência de calor a pressão (P_{ev}) e a temperatura (T_{ev}), desde vapor úmido no estado 4 até atingir o estado de vapor saturado seco ($X=1$). O calor transferido ao refrigerante, no evaporador não modifica a sua temperatura, mas somente muda o seu estado.

5.2 Ciclo real de compressão a vapor

O ciclo da Figura 5.2 apresenta algumas diferenças com relação ao ciclo padrão, caracterizando-se principalmente pela ineficiência dos processos envolvidos. As principais diferenças residem em perdas de carga, perdas de calor adicionais para o ambiente, atrito relacionado ao escoamento do fluido e às partes moveis e misturas de substancias diferentes do fluido refrigerante (Exemplo lubrificante).

O processo de compressão 1→2 (1→2') inicia-se com vapor superaquecido na sucção do compressor, é então, comprimido e passa por um processo de aumento ou diminuição de entropia (1→2), com irreversibilidade e transferência de calor para o refrigerante. Ocorre, ainda, entre os estados 1 e 2 (1 e 2') perdas de carga na válvula de sucção (ponto 1). A Figura 5.2 apresenta o diagrama do ciclo real de compressão a vapor onde é mostrado o processo de compressão (1→2'), que se inicia como vapor super aquecido na sucção (ΔP_s) e de descarga (ΔP_d) do compressor, além do superaquecimento (ΔT_s) e sub-resfriamento (ΔT_d) para este mesmos pontos.

O processo de condensação se inicia com o vapor superaquecido (ponto 3). Durante este processo há perda de carga devido ao escoamento (ΔP_{cd}), no interior do condensador, e posteriormente um grau de sub-resfriamento (que garante a entrada de líquido na válvula de expansão (ponto 5)).

Durante o processo de evaporação aparece mais uma parcela de perda de carga (ΔP_{ev}), que se acentua ao término da evaporação na condição de vapor superaquecido. Nas linhas de refrigerante ocorre transferência de calor (ΔT_{6-7} e ΔT_{8-1}) e de sub-resfriamento (ΔT_{2-3} e ΔT_{4-5}), além das perdas de carga.

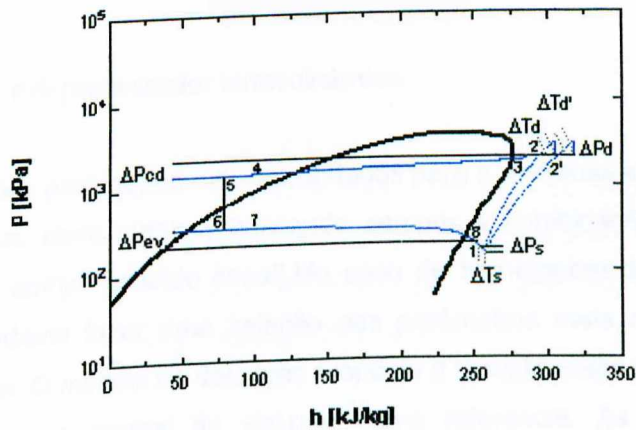


Figura 5.2 Ciclo real de compressão a vapor.

5.3 Modelo de detecção e diagnóstico de falhas

O modelo de detecção e diagnóstico de falhas (DDF) tem como informação básica as grandezas (parâmetros) medidas pelos sensores (Pressão, temperatura, etc), instalados na bancada experimental que realiza o ciclo de compressão de vapor. As grandezas são tratadas matematicamente utilizando um algoritmo pré-processador termodinâmico, Figura 5.3, para obter as propriedades térmicas não mensuráveis como: entalpia, entropia, COP , UA , \dot{m}_{ref} , etc, e acompanhar a evolução das grandezas e propriedades no tempo. As propriedades e parâmetros obtidos são usados para gerar uma base de dados, que identificam inicialmente a operação normal do sistema em regime estável e em condições de ensaio definidas pelos fabricantes do compressor. Esta base de dados é usada para treinar a ANN, em operação chamada de normal (sem-falha).



Figura 5.3 Esquema do Pré-processador termodinâmico.

A identificação dos parâmetros mais apropriados para o treinamento da ANN é baseada no conhecimento físico dos componentes, buscando sempre a combinação de parâmetros mais comportada (Exemplo: comportamento linear). No caso de treinamento da ANN, para identificar uma falha, é recomendável fazer uma seleção dos parâmetros mais sensíveis a falhas bem determinadas e isoladas. O modelo de detecção proposto e testado nesta dissertação (Figura 5.4), usa a operação estável e normal do sistema como referencia. As falhas (artificiais) são introduzidas e a comparação entre a operação estável e normal e a defeituosa, é feita observando a evolução temporal dos parâmetros mais sensíveis, encontrados durante a análise da operação com falhas. O diagnóstico é feito comparando o comportamento dos parâmetros mais sensíveis, tanto para operação normal quanto para com falha.

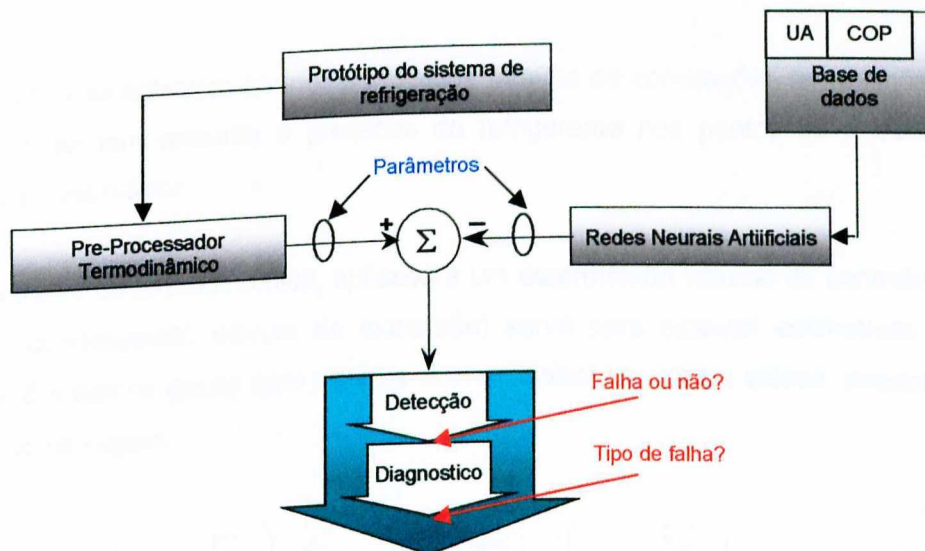


Figura 5.4. Diagrama da metodologia de detecção e diagnóstico de falha.

5.4 Identificação física dos parâmetros

O coeficiente de operação (COP) é usualmente definido como sendo a relação entre o efeito útil desejado e o trabalho necessário para obter esse efeito útil que pode ser refrigeração ou aquecimento (bomba de calor).

$$COP = \frac{\text{Efeito útil}}{\text{Trabalho necessário}}$$

No caso do sistema operando em ciclo ideal (ciclo padrão), são usados dois parâmetros bem conhecidos da literatura consultada: o coeficiente de performance e o rendimento isentrópico do compressor. Estes são obtidos utilizando as entalpias dos estados mostrados no diagrama pressão-entalpia (Figura 5.1), e podem ser escritos como:

$$cop = \frac{h_1 - h_4}{h_2 - h_1} \quad (5.1)$$

$$\eta_{\text{isentrópico}} = \frac{h_{2s} - h_1}{h_2 - h_1} \quad (5.2)$$

Os valores das entalpias podem ser obtidos através de correlações termodinâmicas, quando se conhecem as temperaturas e pressões do refrigerante nos pontos de entrada e saída do compressor e evaporador.

A primeira lei da termodinâmica, aplicada a um determinado volume de controle (compressor, evaporador, condensador, válvula de expansão) serve para calcular estimativas de vazão de refrigerante. Em termos gerais para sistema com múltiplas entradas e saídas, podemos escrever a primeira lei como segue:

$$\sum_{i=1}^n \dot{Q}_i + \sum_{i=1}^n \dot{m}_{i,su} \left(h_{su,i} + \frac{V_{su,i}^2}{2} \right) + \sum_{i=1}^n W_i = \frac{dU}{d\tau} + \sum_{i=1}^n \dot{m}_{i,ex} \left(h_{ex,i} + \frac{V_{ex,i}^2}{2} \right) \quad (5.3)$$

A Equação 5.3 pode ser simplificada desprezando-se as variações de energia cinética e potencial e assumindo também, condições de regime permanente no volume de controle, ou seja: $dU/d\tau = 0$.

Assim a expressão simplificada é dada pela Equação 5.4.

$$\sum_{i=1}^n \dot{Q}_i + \sum_{i=1}^n (\dot{m}\Delta h)_i + \sum_{i=1}^n \dot{W}_i = 0 \quad (5.4)$$

Aplicando a equação 5.4 a cada um dos fluidos que trocam calor nos trocadores de calor, obtêm-se as equações para o fluxo de calor.

$$\text{Água: } \dot{Q} = \dot{m}_w C_{p_w} (T_{w,ex} - T_{w,su}) \quad (5.5)$$

$$\text{Refrigerante: } \dot{Q} = \dot{m}_{ref} (h_{ref,ex} - h_{ref,su}) \quad (5.6)$$

Sabendo que para trocadores de calor, também podemos utilizar a equação de troca térmica baseada em termos de resistência térmica, conforme equação 5.7.

$$\dot{Q} = UA * LMTD \quad (5.7)$$

Onde: ($LMTD$) é a diferença média logarítmica da temperatura definida para o condensador como:

$$LMTD_{cd} = \frac{(T_{cd} - T_{w,su}) - (T_{cd} - T_{w,ex})}{\ln\left(\frac{(T_{cd} - T_{w,su})}{(T_{cd} - T_{w,ex})}\right)} \quad (5.8)$$

E para o evaporador como:

$$LMTD_{ev} = \frac{(T_{w,su} - T_{ev}) - (T_{w,ex} - T_{ev})}{\ln\left(\frac{(T_{w,su} - T_{ev})}{(T_{w,ex} - T_{ev})}\right)} \quad (5.9)$$

Sendo UA o coeficiente global de troca térmica e \dot{m}_{ref} , T_{cd} , T_{ev} , fluxo de refrigerante, temperatura de condensação e evaporação respectivamente.

Se observarmos o comportamento da temperatura nos trocadores de calor, Figura 5.4, é possível identificar parâmetros para o condensador $APPRc$ – diferença entre T_{cd} e $T_{w,ex,cd}$ para o evaporador, $AFPRE$ – diferença entre $T_{w,ex,ev}$ e T_{ev} , e também definir $DTWCD$ e $DTWEV$ como a diferença absoluta entre as temperaturas da água na entrada e saída do condensador ou do evaporador.

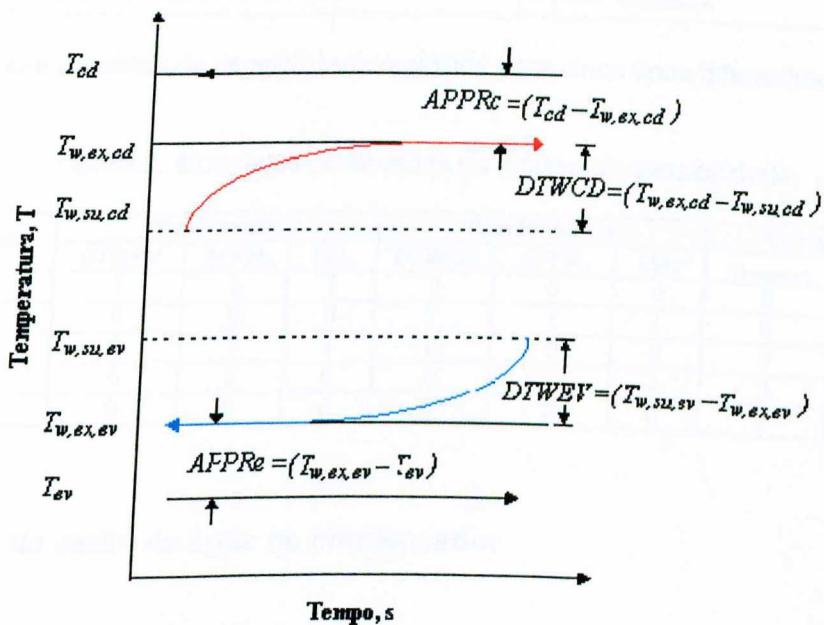


Figura 5.4 Parâmetros identificados pela evolução da temperatura no condensador e no evaporador.

5.4.1 Sensibilidade de parâmetros

Parte da informação sobre a sensibilidade dos parâmetros foi encontrada na literatura (McIntosh I. B., 1999) e outras são frutos de nossas próprias experiências. Cada falha foi estudada independentemente e comparada à operação normal (sem-falha) a través de dados experimentais. Observamos que alguns parâmetros são mais sensíveis a determinadas falhas que a outra.

A Tabela 5.1 mostra os parâmetros sensíveis para as falhas específicas

Tabela 5.1. Parâmetros sensíveis.

Tipo de falha	Parâmetros sensíveis
(1) Redução da vazão de água no evaporador	$UA_e, APPR_e, DTWEV$
(2) Redução da vazão de água no condensador	$UA_c, APPR_c, DTWCD$
(3) Obstrução no tubo do evaporador	UA_e e $APPR_e$
(4) Obstrução no tubo do condensador	UA_c e $APPR_c$
(5) Falha interna no compressor	P, COP e $\eta_{isentrópico}$

A Tabela 2 mostra a análise de sensibilidade realizada para cinco tipos diferentes de falhas.

Tabela 2. Mudanças observadas na análise de sensibilidade.

Tipode falha	Evaporador			Condensador			Compressor	
	$DTWEV$	$APPR_e$	UA_e	$DTWCD$	$APPR_c$	UA_c	$\eta_{isentrópico}$	η_{motor}
(1)	↑	↓	↑	0	0	0	0	0
(2)	0	0	0	↑	↑	↓	0	0
(3)	0	↑	↓	0	0	0	0	0
(4)	0	0	0	0	↑	↓	0	0
(5)	0	0	0	0	0	0	↓	0

5.4.2 Redução da vazão de água no condensador

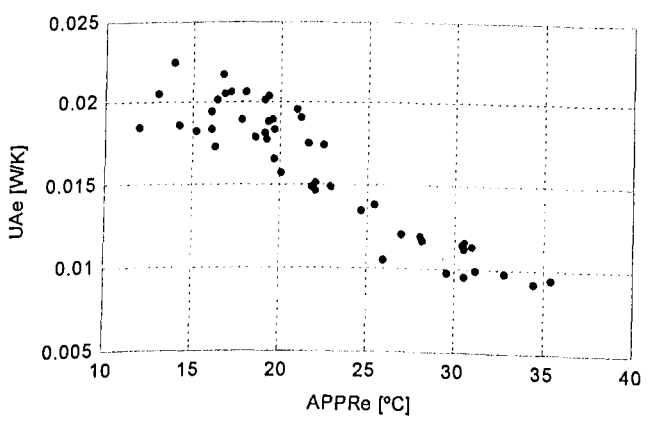
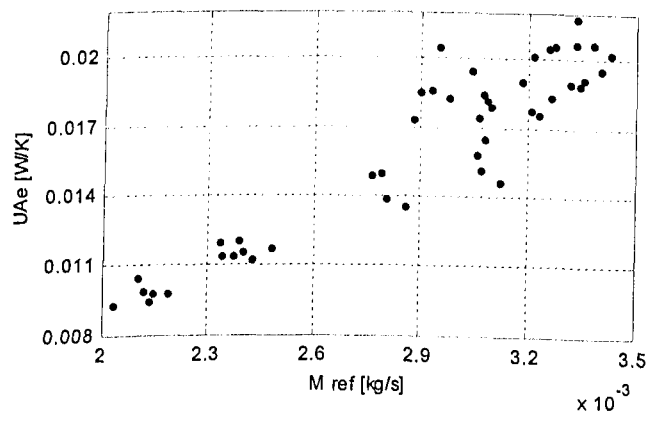
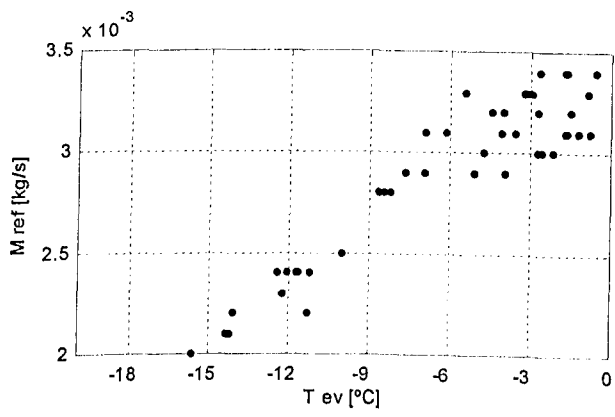
Devido à redução da vazão (causada artificialmente), ocorre uma diminuição no coeficiente de transferência de calor da água que passa através dos tubos do condensador e conseqüentemente, UA_c é afetado. Também se observou um aumento na temperatura e pressão de saturação e conseqüentemente, a temperatura da água na saída do condensador.

5.4.3 Redução de vazão de água no evaporador

Similar ao condensador, aqui, a redução da vazão da água (causada artificialmente) causa um decréscimo na temperatura de evaporação, na temperatura da água na saída do evaporador e um incremento no UA_e . Observa-se que o parâmetro mais sensível para esta falha é $DTWEV$.

Os parâmetros encontrados tiveram sua sensibilidade analisada (ver tabela 2). Esta análise permitiu observar a combinação dos parâmetros mais apropriados (linearmente exemplo: $m_{ref} \times T_{ev}$

, $m_{ref} \times UA_{ev}$ e $APPR_{ev} \times UA_{ev}$ etc.) em operação normal do sistema. As figuras a seguir mostram alguns dos resultados destas combinações.



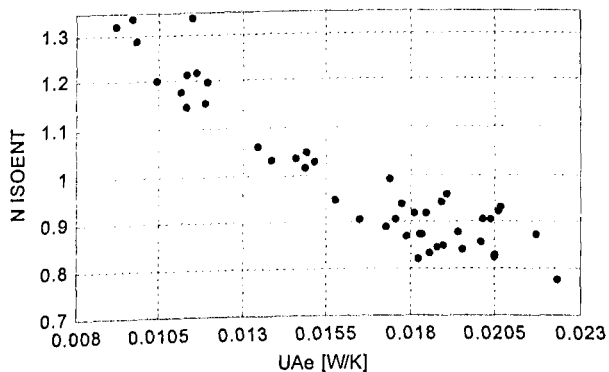
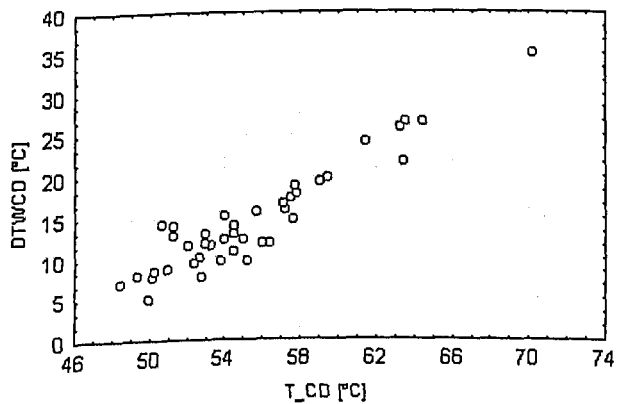
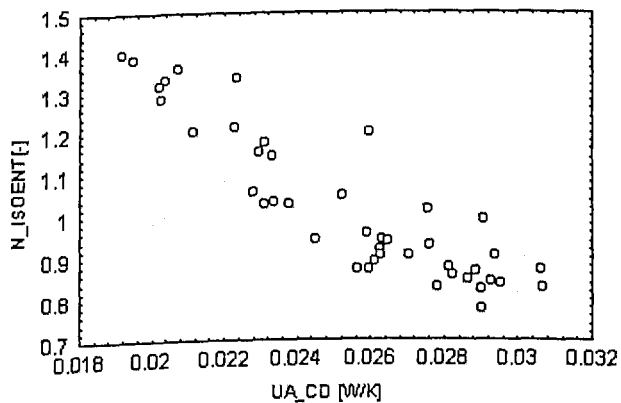


Figura 5.5. Gráficos das comparações dos parâmetros para o evaporador.



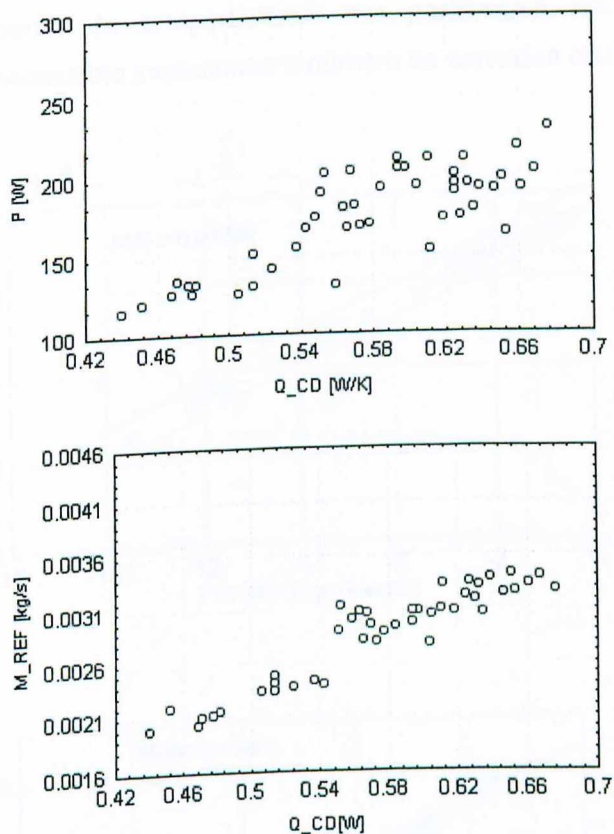


Figura 5.6 Gráficos das comparações dos parâmetros para o condensador.

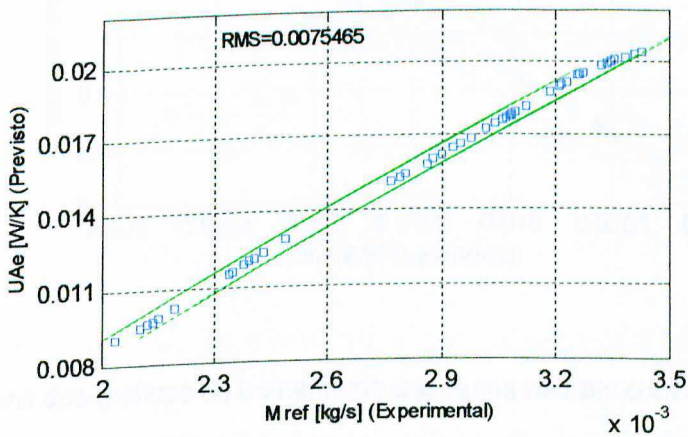
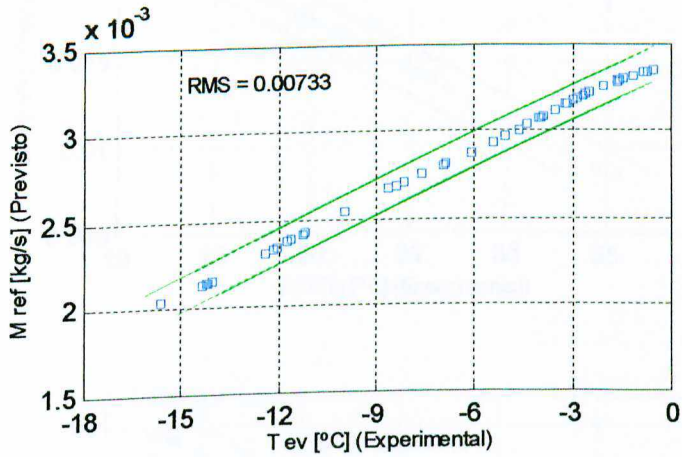
5.5 Estrutura das redes neurais artificiais

A arquitetura da ANN utilizada é uma Perceptron Multi-Layer (MLP) com duas camadas de entrada, duas ou mais camadas ocultas e uma camada da saída. Utiliza o algoritmo "Backpropagation" com a função de ativação sigmóide. O método iterativo do gradiente descendente é usado para minimizar o erro quadrático médio entre a saída desejada e a saída real

5.5.1 Treinamento das redes neurais para o funcionamento normal

As Figuras a seguir mostram resultados do treinamento da ANN em operação normal (foram usados dados experimentais) enquanto as linhas pontilhadas mostram o threshold. A

arquitetura da ANN depende do comportamento dos parâmetros. Se o comportamento for complexo (não linear), é necessário incrementar o número de camadas ocultas.



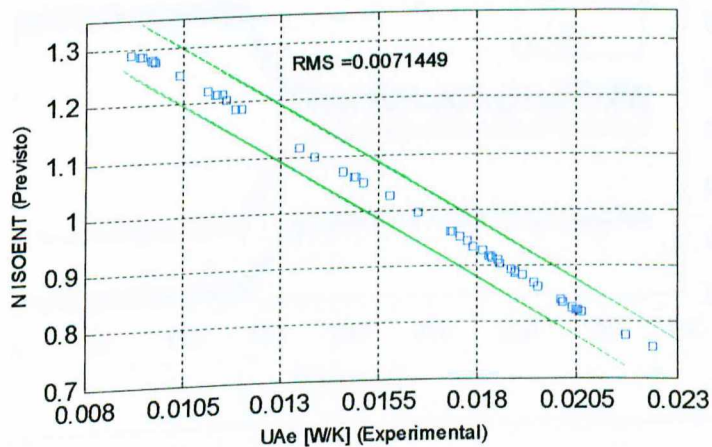
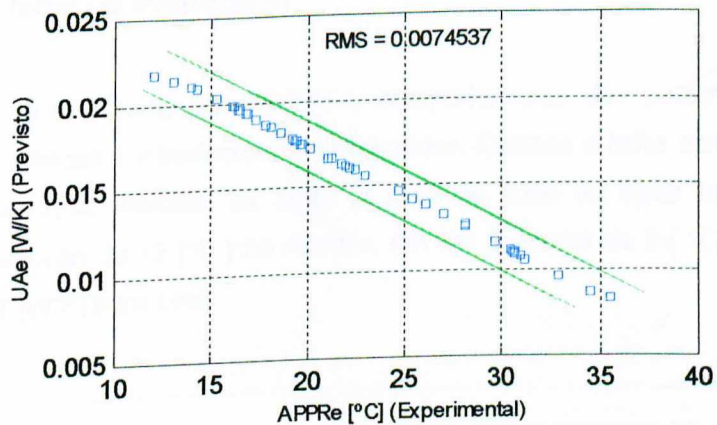


Figura 5.7 Alguns dos gráficos do treinamento das redes neurais com comportamento linear.

5.6 Detecção de falha utilizando esta metodologia

As falhas artificiais foram introduzidas após o sistema estar operando em regime estável e normal, para se fazer as devidas comparações com a base de dados gerada quando do funcionamento normal (sem-falha).

5.6.1 Análise de falha no evaporador

As Figuras (5.8) e (5.9) apresentam o comportamento dos parâmetros mais sensíveis quando uma falha artificial é introduzida no evaporador. Quando a falha consistiu numa mudança redução rápida na vazão mássica da água (a taxa de fluxo da água foi reduzida em 54%). Observa-se uma redução de 12 [°C] no *APPre*, em um aumento de 9 [°C] em *DTWEV*, e num aumento de 0.0307 [W/K] para *UAe*.

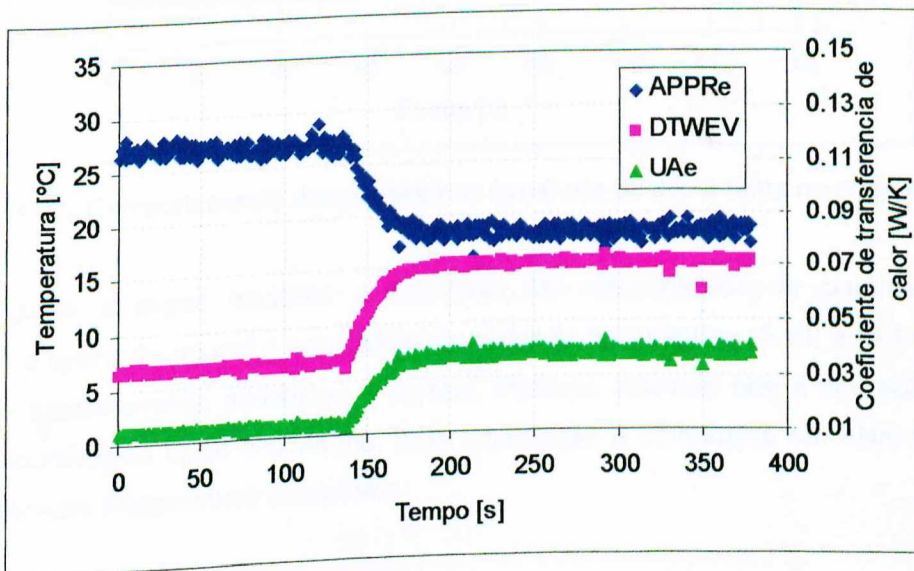


Figura 5.8 Comportamento dos parâmetros sensíveis devido à falha no evaporador

Pode-se observar que até 140 segundos, a operação do sistema foi normal e estável, mas após introduzir a falha, observa-se mudança rápida dos parâmetros mais sensíveis previamente selecionados.

Na Figura 5.9 a falha introduzida é similar à anterior (a vazão de água foi reduzida em 90%), e observa-se uma redução de 22 [°C] no *APPre*, em um aumento de 20 [°C] em *DTWEV*, e um aumento de 0.08551 [W/K] para *UAe*. Portanto, observa-se que até 60 segundos a operação do sistema é normal e estável. Após reduzir a vazão da água, ocorre uma mudança significativa dos parâmetros mais sensíveis antes definidos.

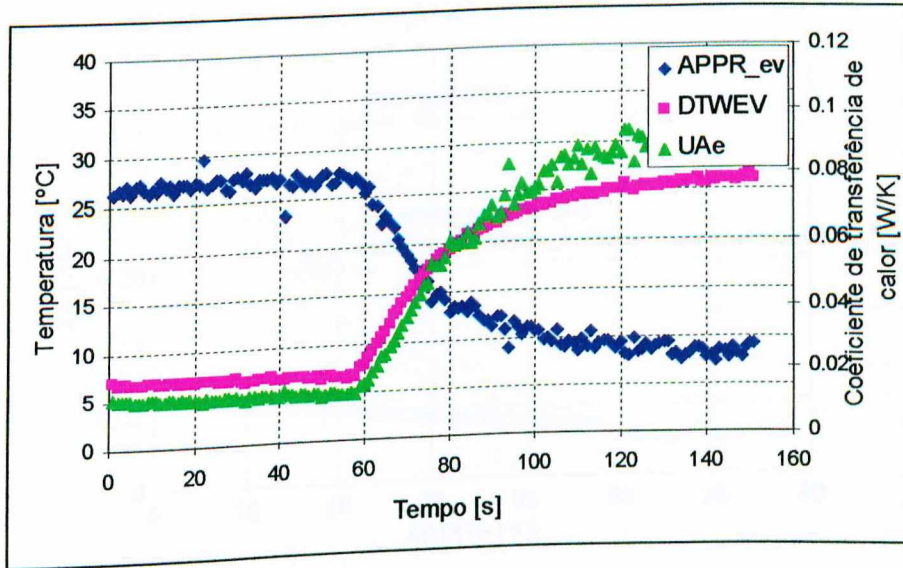


Figura 5.9 Comportamento dos parâmetros sensíveis devido à falha no evaporador

As figuras a seguir mostram o resultado das combinações de parâmetros quando comparamos a operação normal e com falhas do sistema. Nas Figuras (5.10) e (5.11) observa-se claramente o funcionamento defeituoso e normal. Pode-se observar que a operação defeituosa tem um comportamento típico e uniforme. Esta informação é introduzida em ANN para treiná-la para a identificação (diagnóstico) desta falha.

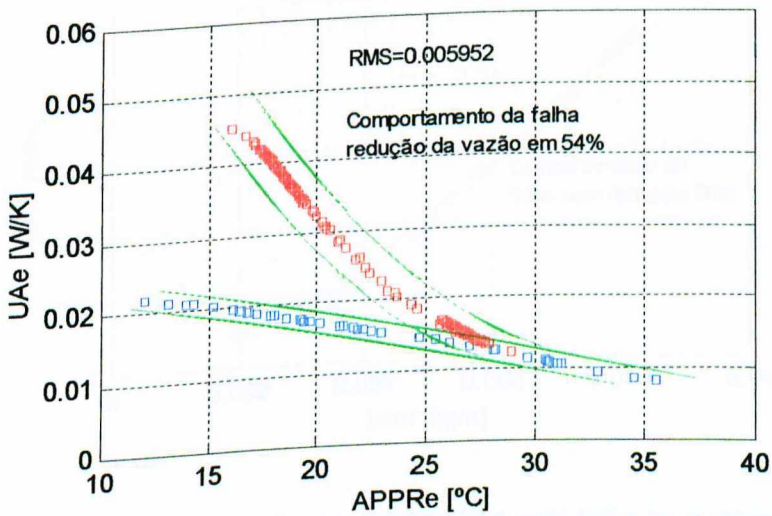


Figura 5.10 Combinação dos parâmetros para falha no evaporador

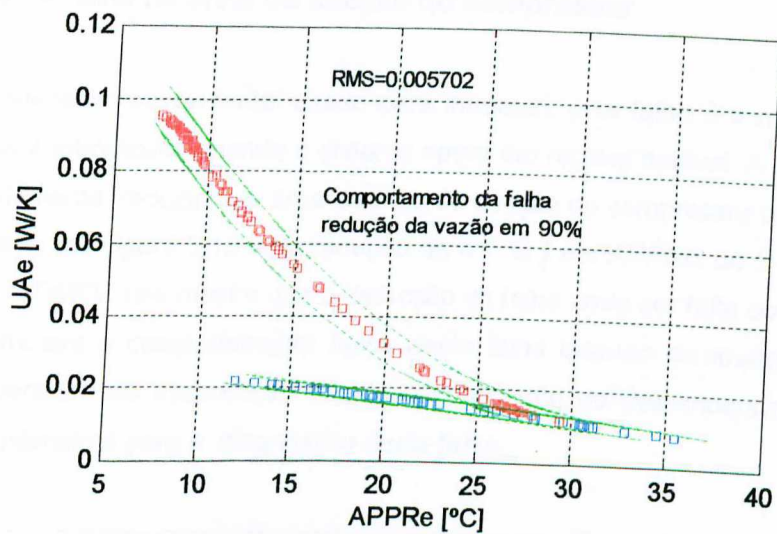


Figura 5.11 Combinação dos parâmetros para falha no evaporador

A Figura 5.12 apresenta outra combinação dos parâmetros para a mesma falha. Não se recomenda usar esta combinação para o diagnóstico da falha, porque o comportamento não é uniforme nem evidente e difícil de ser identificado.

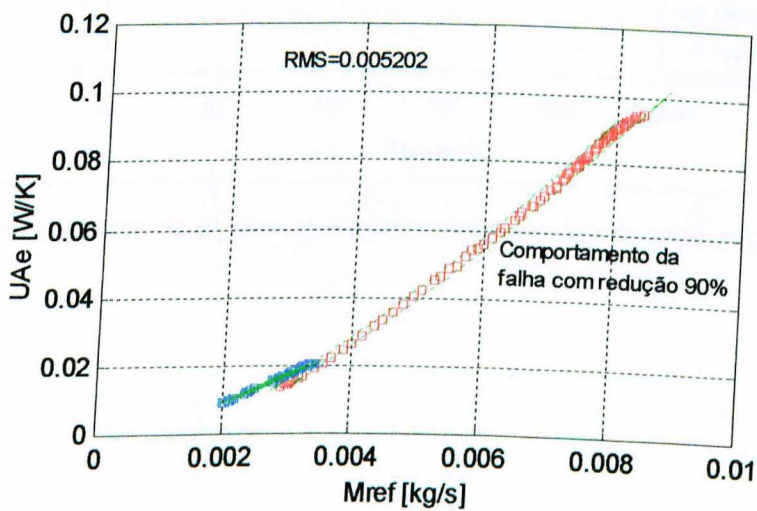


Figura 5.12 combinação dos parâmetros para falha no evaporador

5.6.2 Análise de falha na linha de sucção do compressor

O procedimento experimental usado para introduzir esta falha é o mesmo usado no item anterior. A falha é introduzida quando o sistema opera em regime estável. A vazão do refrigerante é mudada rapidamente, reduzindo a área da linha de sucção do compressor em 90%.

Observa-se na Figura 5.13 uma redução de 4 [°C] em $APPR_c$, de 2 [°C] em $DTWCD$ e de 25 [°C] em $DTWEV$. Isto mostra que a detecção de falha pode ser feita com estes parâmetros. A Figura 5.16 mostra o comportamento típico desta falha quando comparada com a operação normal. A temperatura da evaporação (T_{ev}) e o coeficiente do desempenho (COP) podem ser usados como parâmetros para o diagnóstico desta falha.

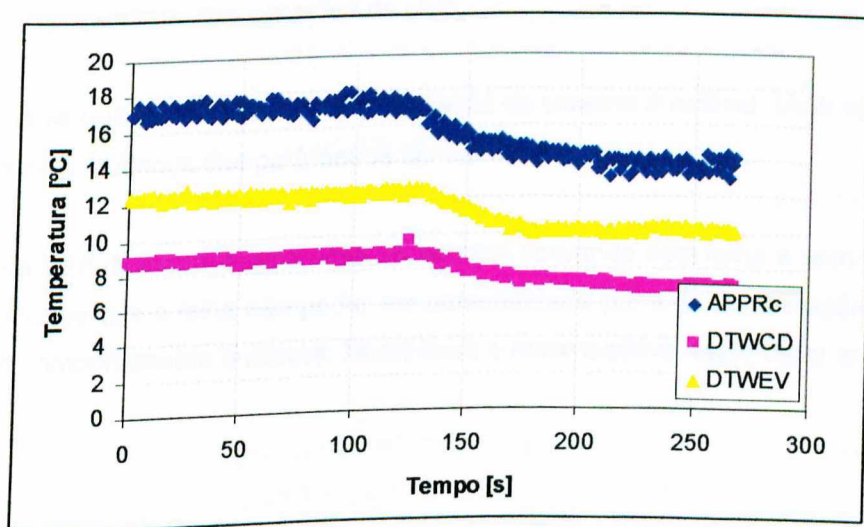


Figura 5.13 Comportamento de alguns parâmetros sensíveis devido a falha na linha de sucção do compressor.

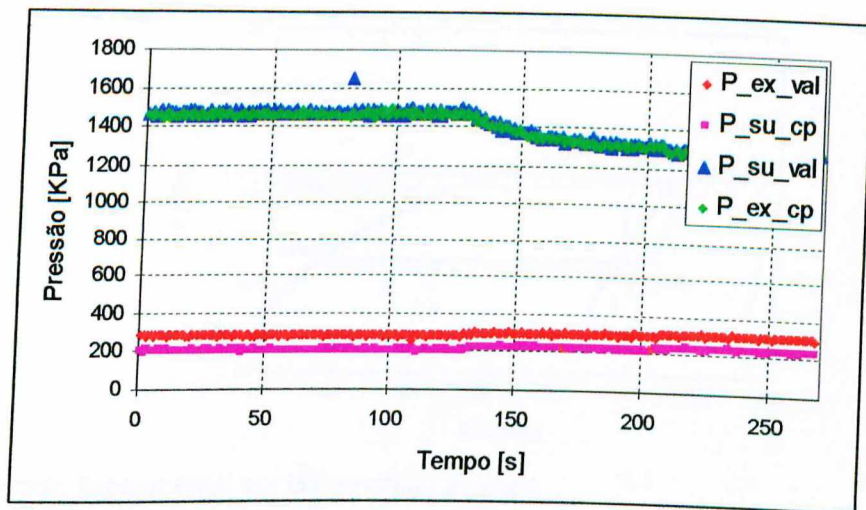


Figura 5.14 Comportamento das pressões do ciclo.

Observa-se que até 130 segundos a operação do sistema é estável. Uma vez introduzida a falha, observa-se a mudança dos parâmetros sensíveis.

A Figura 5.15 mostra a comparação do sistema operando com falha e sem falha (operação normal). Observa-se que a falha não poder ser caracterizada por esta combinação de parâmetros por não ter um comportamento uniforme. Neste caso é recomendável fazer outra análise.

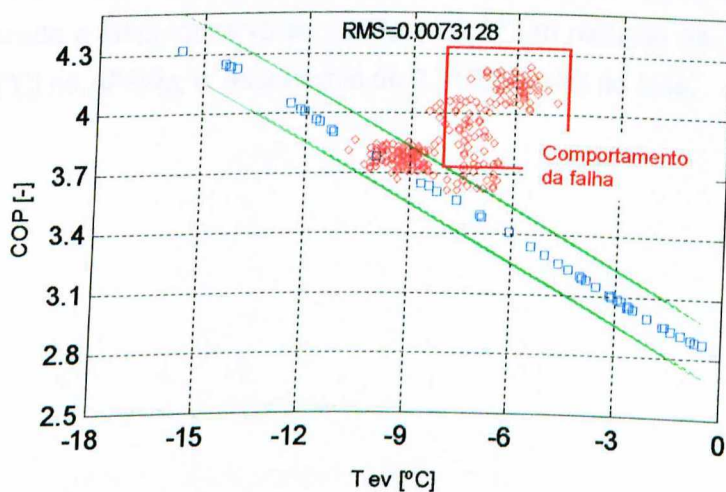


Figura 5.15 Comparação de T_{ev} e COP para o diagnóstico de falha no compressor

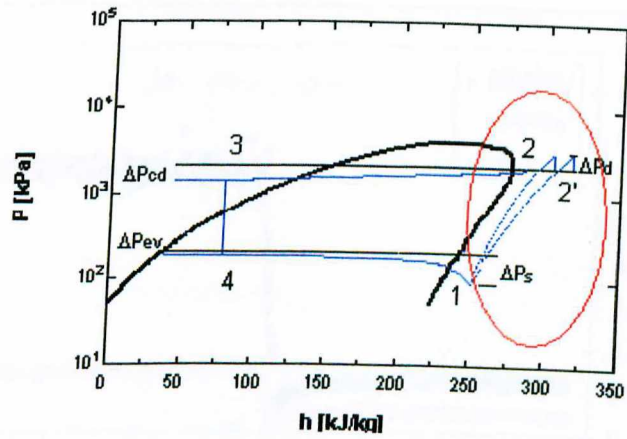


Figura 5.16 do ciclo experimental por compressão a vapor.

Na análise desta falha não foi possível caracterizar o seu comportamento típico por falta de parâmetros de análise. Observando a Figura 5.16 verifica-se que com a redução da vazão de refrigerante no duto de sucção do compressor, tem-se um incremento no (ΔP_s) o que provoca diminuição da eficiência e do consumo de potencia do compressor, efeito que não foi levado em consideração por falta de instrumentação no sistema.

5.6.3 Análise da falha na válvula de expansão

O procedimento é o mesmo realizado nas duas falhas anteriores. A falha é introduzida depois que o sistema entra em regime estável, a falha foi o fechamento da válvula de expansão. Uma vês introduzida a falha, observa-se na Figura 5.17 um redução de 7 [°C] no $DTWEV$, e um aumento de 27 [°C] no $APPRE$, e decréscimo de 0.01035 [W/K] no UAe .

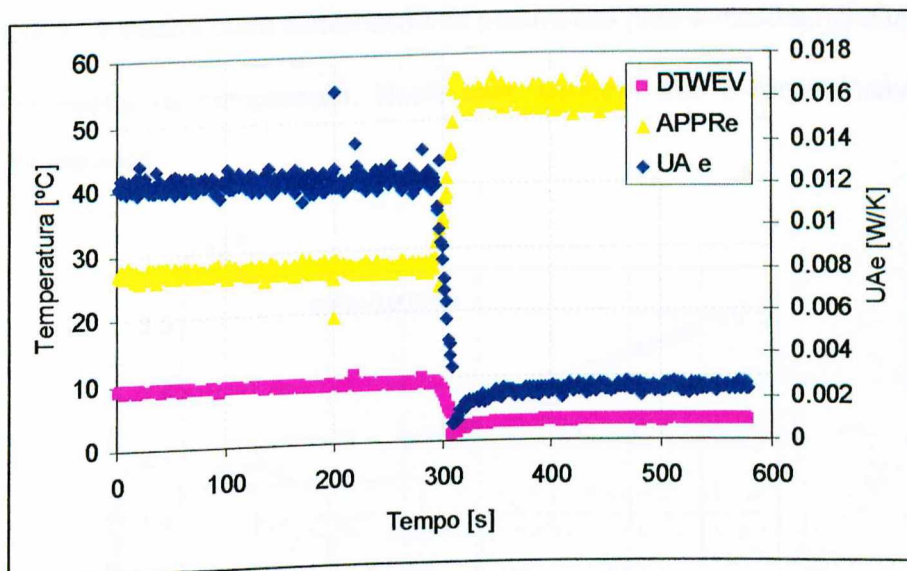


Figura 5.17 Comportamento dos parâmetros sensíveis com falha na válvula de expansão.

A Figura 5.18 mostra os resultados da comparação da operação normal e com falha. Observa-se um decréscimo da T_{ev} e uma diminuição em \dot{m}_{ref} através de um comportamento linear. Não é recomendável utilizar estes parâmetros para caracterizar esta falha porque no gráfico pode-se observar que a operação normal e com falha apresenta um comportamento similar.

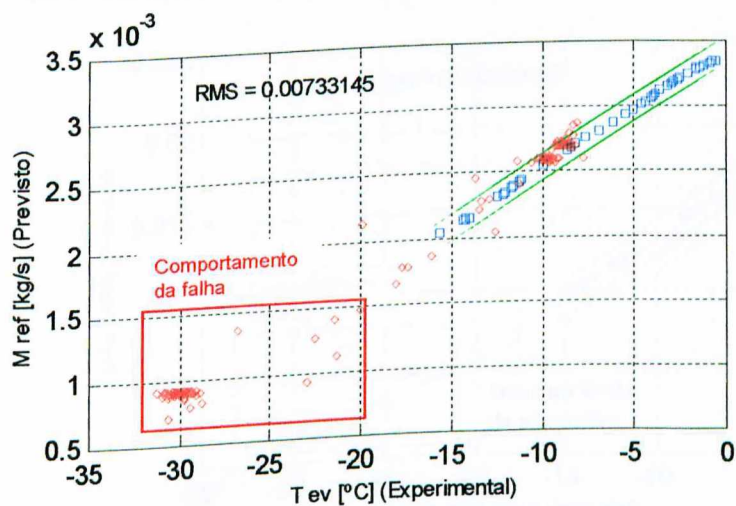


Figura 5.18 Combinação de T_{ev} experimental com m_{ref} previsto.

A Figura 5.19 mostra outra combinação de parâmetros para a mesma falha usando UA_e e m_{ref} como parâmetros de comparação. Neste caso também não é recomendável utilizar os parâmetros comparados.

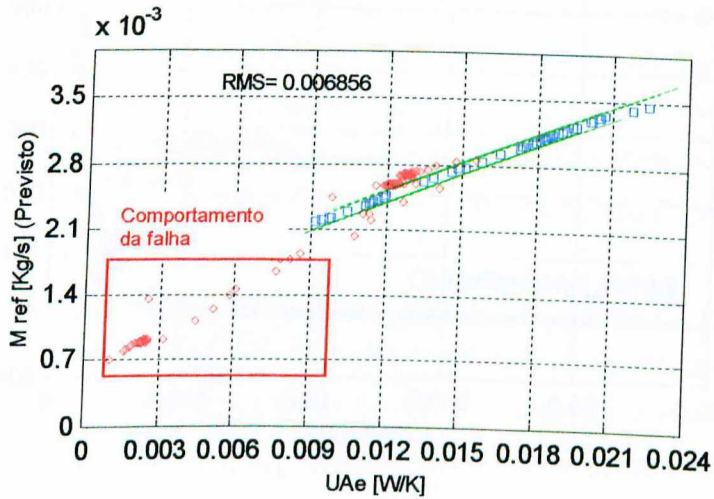


Figura 5.19 Combinação de T_{ev} experimental com UA_e previsto.

Na Figura 5.20 pode-se observar um comportamento quase uniforme para esta falha, o que também não é recomendável, conforme mencionado nas Figuras 5.17 e 5.18.

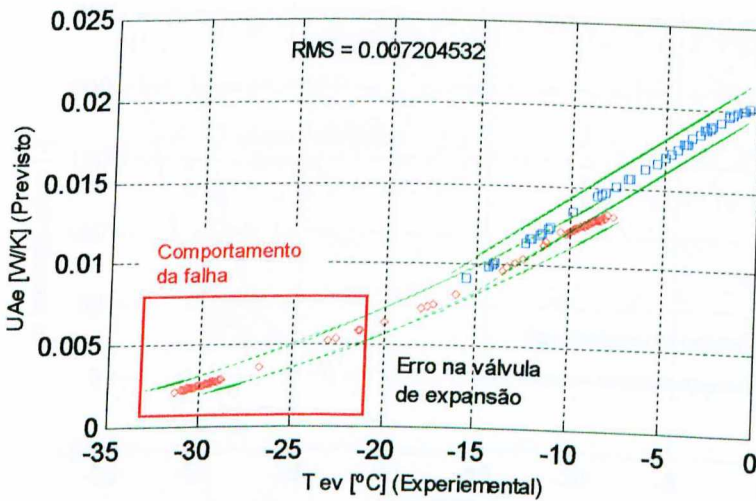


Figura 5.20 Combinação de T_{ev} experimental com UA_e previsto.

A Figura 5.21 mostra uma combinação onde pode-se observar um comportamento uniforme para esta falha que é utilizado para caracterizar falha.

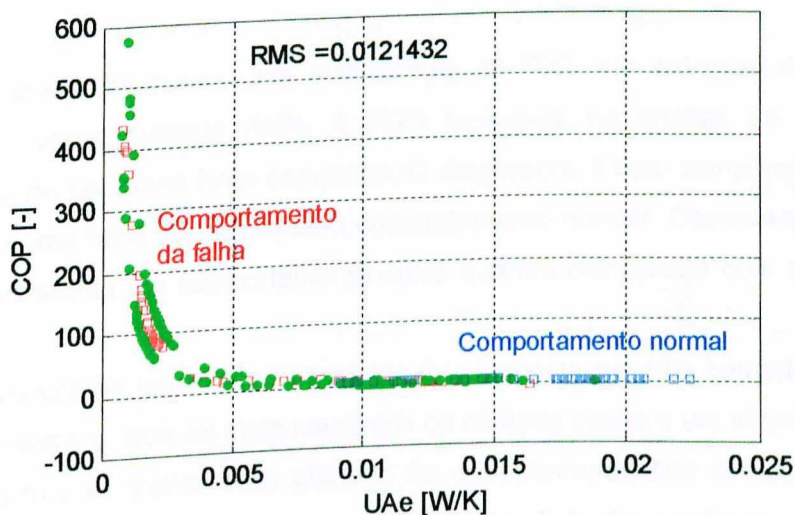


Figura 5.21 Combinação de UA_e experimental com COP.

A Figura 5.22 mostra uma combinação onde pode-se observar um comportamento uniforme similar a figura anterior com diferentes parâmetros. Esta combinação é utilizado para caracterizar falha.

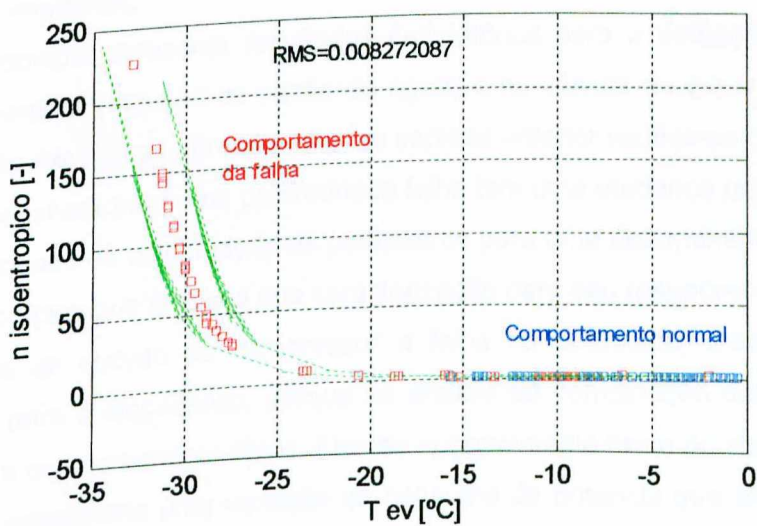


Figura 5.22 Combinação de T_{ev} experimental com $n_{\text{iseotropico}}$.

CAPÍTULO VI

Conclusões

Este trabalho apresenta uma metodologia de FDD, em sistemas de refrigeração por compressão a vapor, usando ANN. A FDD baseia-se na análise de sensibilidade dos parâmetros físicos para uma falha específica. O diagnóstico é feito comparando os parâmetros sensíveis para uma falha específica com comportamento normal. Observa-se que a operação com falha apresenta um comportamento típico quando comparada com a operação normal do sistema.

A arquitetura da rede neural escolhida foi composta por três camadas: dois neurônios na camada de entrada, dois ou mais neurônios na camada oculta e um neurônio na camada de saída. O algoritmo de treinamento utilizado foi o *backpropagation* através da utilização da função de ativação sigmóide, usado o método iterativo do gradiente descendente para minimizar o erro quadrático médio entre a saída desejada e a saída real. Durante o treinamento da rede o erro médio quadrático estipulado foi atingido, tanto para treinamento em funcionamento normal como no treinamento em operação com falha. Para o treinamento da rede foi necessária uma base de dados experimentais em regime permanente do sistema, onde a escolha dos pontos normais de funcionamento do sistema foi feita em base dos catálogos do fabricante do compressor.

A metodologia apresenta resultados satisfatórios para a detecção e diagnóstico das falhas no evaporador (redução da vazão de água) e na válvula de expansão (fechamento da válvula). Através das figuras apresentadas no capítulo anterior verifica-se que o comportamento das variáveis sensíveis para uma determinada falha tem uma mudança rápida. Isso implica que a escolha apropriada da combinação de parâmetros para uma determinada falha apresenta um comportamento típico que facilita a sua caracterização para seu respectivo diagnóstico.

Na linha de sucção do compressor a falha foi detectada, mas não foi possível a caracterização para o diagnóstico, porque na análise da combinação dos parâmetros não foi apresentado um comportamento típico. Através do conhecimento físico do sistema pode-se prever que esta falha ocasionaria uma variação no consumo de potência que afetaria a eficiência do compressor fato que não foi possível mostrar experimentalmente devido à falta de instrumentação adequada. Acredita-se que com o uso de instrumentação adequada que permita medir a mudança temporal da potência consumida em tempo real, conseguiríamos outros parâmetros de análise, o que possibilitaria caracterizar a falha.

A grande vantagem deste método é a facilidade com que as falhas são detectadas pois o usuário não necessita conhecer a física do sistema, pois o método analisa por separado cada um dos componentes do sistema.

Melhoras na instrumentação da bancada experimental permitiriam a análise das falhas, que nesta dissertação não foi possível diagnosticar. Assim, propõe-se algumas alterações na bancada:

- Implantar sensores pressão diferencial antes da entrada linha de sucção do compressor, para melhor avaliar a perda de pressão neste trecho.
- Usar equipamento e software que permita medir a evolução temporal da potência consumida pelo compressor.
- Um calorímetro no compressor para avaliar com precisão as perdas de calor para o ambiente.

Futuros desdobramentos são: generalizar a metodologia proposta, incremento da base de dados para análise de outros tipos de falhas. Fazer comparações com diferentes métodos de detecção de falha relatados na literatura para este tipo de sistema, e analisar as vantagens e desvantagens do método proposto do ponto de vista de implantação, confiabilidade e desempenho.

REFERÊNCIAS BIBLIOGRÁFICAS

ASHRAE. 1993. ASHRAE Handbook of Fundamentals, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc., Atlanta.

Antsaklis, P.J. "Neural Networks in Control Systems." *IEEE Control Systems*, v. 12, n. 2, p. 8-10, April, 1992.

Braun, J.E. 1996. "Thresholds for Fault Detection, Diagnosis and Evaluation." IEA Annex 25, Building Optimization and Fault Diagnosis Source Book, Technical Research Center of Finland, pp.277-282.

Braun, J.E. and M.S. Breuker. 1997. "Demonstration of a Statistical, Rule-Based Fault Detection and Diagnostic Method on a Rooftop Air-Conditioning Unit." *CLIMA 2000 Conference Proceedings*: CD-ROM.

Braun, J.E. and T.M. Rossi. 1997. "A Statistical, Rule-Based Fault Detection and Diagnostic Method for Vapor Compression Air Conditioners." *HVAC&R Research*, Vol.3, No.1, pp.19-37.

Carvajal, F.A. 2000 "Controle Digital de Válvula de Expansão Usando Lógica Difusa." Dissertação FEMEC-UFU.

Dexter A. L. and M. Benouarets. 1996a. "Transient Detection." *IEA Annex Final Report*.

Grimmelius, H.T., J.K. Woud, and G. Been. 1995. "On-line Failure Diagnosis for Compression Refrigeration Plants." *Rev. Int. Froid* 18(1): pp. 31-41.

Mammerstron D., "Neural Networks at Work", *IEEE Spectrum Mag.*, p. 26-32, June 1993a.

Harunori Yoshida, Tatsuhiro Iwami, Hideki Yuzawa, Masami Suzuki., 1996 "Typical Faults of Air-Conditioning Systems and Fault Detection by ARX Model and Extended Kalman Filter", ASHRAE AT-96-3-5.

Henk C. Peitsman, Vicent E. Bakker., 1996 "Application of Black-Box Models to HVAC Systems for Fault Detection", ASHRAE AT-96-6-3.

Herzog, P. and L. Lavine. 1992. "Identification and quantification of the impact of improper operation of midsize Minnesota office buildings on energy use: A seven building case study." Proc. ACEEE 1992 Summer Study on Energy Efficiency in Buildings, pp.3.121-3.129. Washington D.C.: American Council for an Energy Efficient Economy.

Ian B. D. McIntosh., 1999, "A Model-Based Fault Detection and Diagnosis Methodology for HVAC Subsystems" University of Wisconsin – Madison.

Incropera, F.P. and D.P. Dewitt. 1985. Introduction to Heat Transfer. John Wiley & Sons, New York.

João. M. Pimenta, Oscar. H. Mendoza., 1999 "Overview of Fault Detection and Diagnosis Methods Applied to HVAC Systems", COBEM 99, Universidade Federal de Uberlândia.

L. K. Norford., 1998 "Fault Detection and Load Monitoring in Ventilation Systems", Massachusetts Institute of Technology. ASHRAE Transactions.

McIntosh I. B. D., 1999, "A Model-Based Fault Detection and Diagnosis Methodology for HVAC Subsystems" University of Wisconsin – Madison.

Mcdowell Tim, 1992 "Routine Modified, Prepared for inclusion in Trnsys 14.1 7/93"

Maruyama N., M. Benouarets, A. L. Dexter.,1993, "Fuzzy Model-Based Fault Detection", Department of Engineering Science, University of Oxford.

Niwa, H. and N. Nakahara. 1996. "Fault Simulation of an HVAC System with Thermal Storage Tank." Real Time Simulation of HVAC Systems for BOFD: Technical Papers of Annex 25, pp. 275-287.

Philip Haves, Timonhy I. Salsbury, J. A. Wright., 1996 "Condition Monitoring in HVAC Subsystems Using First Principles Models", ASHRAE AT-96-3-1.

PROCEL, (<http://www.eletronbras.gov.br/procel/>), Basil, 2000

- Lee, W.Y., John M. House, Cheol Park, George E. Kelly.,1996 ,“Fault Diagnosis of na Air-Handling Unit Using Artificial Neural Networks” ASHRAE V. 102 Pt. 1, pp 540-549.
- Lee, W.Y., Cheol Park, George E. Kelly., 1996, “Fault Detection in an Air-Handling Unit Using Residual and Recursive Parameter Indentification Methods” ASHRAE V. 102 Pt. 1.
- Lee, W.Y., J.M. House, and D.R. Shin. 1997. “Fault Diagnosis and Temperature Sensor Recovery for an Air-Handling Unit.” ASHRAE Transactions, 103(1).
- Lee, W.Y., John M. House, Cheol Park, Dong Ryul Shin.,1999,“Classification Thechniques for fault Detection and Diagnosis of an Air-Handling Unit Using” ASHRAE CH-99-18-5.
- Li, X., J.C. Visier, and H. Vaezi-Nejad. 1996. “A Neural Network Prototype for Fault Detection and Diagnosis of Heating Systems.” *ASHRAE Transactions*. 102(1).
- Stylianou, M. and D. Nikanpour. 1996. “Performance Monitoring, Fault Detection and Diagnosis of Reciprocating Chillers.” ASHRAE Transactions, 102(1).
- Touretzky, D. S., Pormeleau, D.A. Whast's Hidden in the Hidden Layers?
- Yamada, K. and K. Kamimura. 1996. “Fault Detection of Thermal Storage System by Expert System using Fuzzy Abduction.” *Real Time Simulation of HVAC Systems for BOFD: Technical Papers of Annex 25*, pp.289-296.

ANEXO I

As tabelas a seguir apresentam os cálculos das propriedades térmicas realizados com os dados obtidos da bancada de ensaios.

Tabela 1. Cálculo da vazão do refrigerante nos trocadores de calor.

Fecha	Hora	Potencia [W]	$\dot{m}_{w,cd}$ [kg/s]	$\dot{m}_{w,ev}$ [kg/s]	T_{cd} [°C]	T_{ev} [°C]	$\delta\text{Erro } \dot{m}_{ref}$ [%]	$\dot{m}_{ref,cd}$ [kg.s]	$\dot{m}_{ref,ev}$ [°C]
8/7/2002	13:05	300	0.004917	0.008208	55.73465	-3.8617	13.1	0.003103	0.002697
2/8/2002	11:50	320	0.016167	0.009778	57.1547	-1.66032	8.382	0.00358	0.00328
2/8/2002	11:00	310	0.014278	0.009806	55.7097	-1.62259	9.064	0.003543	0.003222
2/8/2002	10:35	300	0.007028	0.00975	51.23791	-2.51209	10.3	0.003112	0.002791
14/8/2002	11:35	230	0.01008	0.01852	48.44064	-12.0523	15.46	0.002592	0.002191
14/8/2002	12:15	250	0.01008	0.01852	50.15392	-8.613	14.07	0.002976	0.002557
14/8/2002	12:40	270	0.01008	0.01852	50.94161	-3.19353	13.71	0.003515	0.003033
14/8/2002	15:15	290	0.01075	0.01848	50.2019	-3.14807	8.712	0.003486	0.003182
15/8/2002	10:50	310	0.022146	0.008667	57.43934	-2.61523	5.678	0.003455	0.003259
15/8/2002	11:15	305	0.014625	0.008854	57.5223	-2.86233	8.898	0.003474	0.003165
15/8/2002	11:35	300	0.010188	0.008854	57.08683	-3.13793	0.6513	0.003348	0.003326
16/8/2002	17:55	240	0.011667	0.009292	52.90076	-12.1644	6.072	0.00241	0.002263
16/8/2002	14:30	240	0.017292	0.009479	54.42152	-11.5915	8.993	0.002517	0.002291
16/8/2002	19:20	295	0.008958	0.009333	50.61789	-2.08419	-1.69	0.002992	0.003043
17/8/2002	16:10	270	0.011854	0.019438	50.26933	-5.42704	3.719	0.003323	0.003199
20/8/2002	10:30	230	0.014979	0.012042	52.67057	-12.4149	8.213	0.002445	0.002244
20/8/2002	11:05	280	0.009229	0.012042	54.92189	-6.08243	5.188	0.003167	0.003003
20/8/2002	11:50	300	0.009229	0.012042	53.98303	-2.64679	8.075	0.003352	0.003082
20/8/2002	14:25	225	0.013729	0.012042	53.73007	-14.0272	11.1	0.002275	0.002022
20/8/2002	15:25	220	0.010417	0.012042	53.53592	-14.228	10.8	0.002241	0.001999
20/8/2002	16:00	220	0.006063	0.012042	52.30933	-14.3435	10.62	0.002221	0.001985
20/8/2002	16:50	265	0.006063	0.012042	53.22149	-7.62944	6.014	0.00297	0.002791
20/8/2002	17:40	280	0.006063	0.012042	52.0442	-5.04018	10.38	0.003093	0.002772
21/8/2002	12:10	305	0.011063	0.012083	54.32037	-1.43366	12.26	0.003394	0.002978
21/8/2002	14:15	300	0.006917	0.012021	52.95818	-2.65631	12.85	0.003181	0.002773
22/8/2002	11:10	300	0.00985	0.010133	57.55362	-3.9647	9.05	0.003361	0.003057
22/8/2002	14:50	250	0.008542	0.004958	63.44552	-11.2562	8.237	0.002288	0.0021
23/8/2002	11:00	220	0.020229	0.014708	52.77611	-14.1875	10.14	0.002251	0.002023
23/8/2002	12:00	315	0.020229	0.014708	54.49786	-0.53336	8.753	0.003564	0.003252
23/8/2002	15:10	310	0.012563	0.013417	55.92504	-0.82714	11.3	0.003545	0.003144
23/8/2002	16:15	240	0.012563	0.013417	55.1512	-11.1603	10.84	0.002567	0.002289
23/8/2002	17:30	295	0.012396	0.013188	56.37181	-4.36927	7.085	0.003351	0.003113
27/8/2002	16:15	210	0.015333	0.022583	49.93757	-15.637	10.98	0.002154	0.001917
28/8/2002	9:35	210	0.010208	0.006958	53.98787	-15.6337	9.636	0.002097	0.001895
28/8/2002	14:50	345	0.021854	0.003771	70.16092	-1.1512	12.73	0.003337	0.002913
28/8/2002	18:25	320	0.01425	0.007604	57.73639	-8.15317	7.798	0.002918	0.002691
28/8/2002	18:50	310	0.01425	0.007604	59.35252	-2.96512	8.008	0.0034	0.003127

28/8/2002	19:25	310	0.008271	0.007604	57.64969	-3.55693	10.98	0.003258	0.0029
29/8/2002	9:45	280	0.008813	0.011208	51.1911	-4.64662	10.91	0.003218	0.002867
29/8/2002	12:30	260	0.008813	0.011208	53.79964	-8.41735	9.635	0.002932	0.002649
29/8/2002	16:00	330	0.014375	0.005125	63.49646	-1.64883	12.75	0.003269	0.002852
29/8/2002	16:50	310	0.014375	0.005083	64.45458	-4.03691	8.514	0.003206	0.002933
29/8/2002	17:25	260	0.014375	0.005083	61.45455	-9.95539	10.02	0.002616	0.002354
29/8/2002	18:00	290	0.014375	0.005083	63.26552	-6.86115	8.698	0.002991	0.002731
30/8/2002	14:30	320	0.012708	0.007333	59.01251	-1.59712	13.85	0.003312	0.002853
31/8/2002	9:30	260	0.014021	0.019292	49.36103	-6.92831	9.324	0.003213	0.002913
31/8/2002	15:45	240	0.016417	0.010542	53.81433	-11.711	8.354	0.002479	0.002272
3/9/2002	10:00	310	0.012417	0.010417	54.49703	-0.74268	14.39	0.003336	0.002856

Tabela 2. Calculo das propriedades térmicas e da diferença de temperatura nos trocadores com o funcionamento estável e sem falha.

COP [-]	$n_{isoentropico}$ [-]	UA_c [W/K]	UA_e [W/K]	\dot{Q}_{cd} [W]	\dot{Q}_{ev} [W]	$\Delta T_{w,cd}$ [°C]	$\Delta T_{w,ev}$ [°C]	$APPR_c$ [°C]	$APPR_e$ [°C]
3.058	0.919	0.02625	0.0185	0.5506	0.3605	16.06244	17.55978	13.96	12.01
3.133	0.906	0.027	0.02016	0.6501	0.4515	15.91821	6.686446	16.99	19.22
3.311	0.9297	0.02757	0.02065	0.6381	0.4457	15.5789	7.472641	16.22	18.06
3.142	0.8226	0.02898	0.02048	0.5695	0.3875	13.98318	13.20027	13.48	13.08
4.443	1.199	0.02594	0.01198	0.5424	0.3743	7.01108	8.888849	17.6	27
3.771	1.015	0.02752	0.01486	0.6032	0.4096	7.797371	9.7295	18.25	22.99
3.236	0.8276	0.03064	0.02051	0.6756	0.4453	8.733236	10.57761	17.97	16.85
3.468	0.8726	0.03059	0.02174	0.661	0.4683	8.562996	10.43017	17.61	16.75
3.233	0.9571	0.02587	0.01912	0.6269	0.4516	17.31564	4.881832	16.6	21.26
3.151	0.9411	0.0263	0.01895	0.6318	0.4369	17.083	7.152312	16.48	19.67
3.086	0.9221	0.02613	0.02064	0.6113	0.4587	16.53024	10.7802	16.1	17.26
3.925	1.153	0.0229	0.01188	0.5052	0.3782	13.01677	7.760348	16.19	28.1
4.467	1.337	0.02228	0.01151	0.5134	0.3818	12.96703	5.285838	17.16	30.6
3.039	0.7748	0.02898	0.02238	0.5587	0.4275	14.33016	11.42304	12.99	13.96
3.498	0.9068	0.02937	0.02041	0.6526	0.4886	8.037997	9.868639	18.44	19.35
4.142	1.215	0.02227	0.01137	0.5137	0.3798	10.21374	6.070732	18.34	30.46
3.153	0.9168	0.02622	0.01813	0.618	0.4448	12.28754	11.54017	17.95	19.22
3.102	0.8553	0.02821	0.02011	0.6245	0.4341	12.41756	11.26267	16.51	16.44
4.375	1.334	0.02033	0.009726	0.4816	0.3485	9.576774	6.077848	19.23	32.88
4.216	1.286	0.02022	0.009821	0.4769	0.3439	9.483321	7.904074	19.17	31.21
3.966	1.203	0.0211	0.01042	0.4706	0.336	9.35822	13.26847	17.95	26.08
3.024	0.89	0.02611	0.01732	0.5774	0.4078	11.48034	16.1052	16.87	16.4
2.975	0.831	0.0278	0.01859	0.5839	0.3917	11.61053	15.46932	15.73	14.28
3.129	0.847	0.02862	0.01897	0.6298	0.4188	12.48044	9.064603	16.35	17.85
3.02	0.8203	0.02901	0.01828	0.5934	0.3885	11.82101	13.45007	15.11	15.23
3.089	0.9367	0.02645	0.01776	0.6246	0.4291	14.75889	10.43256	17	19.33
3.938	1.384	0.01945	0.009705	0.4516	0.3305	21.81311	9.266209	13.99	29.63
4.529	1.361	0.02067	0.009393	0.4791	0.3527	7.799714	4.174412	19.49	35.5
3.177	0.8386	0.02952	0.01954	0.6674	0.4632	10.86597	5.482988	17.61	21.07
3.062	0.8451	0.02925	0.01884	0.6589	0.4406	11.7608	8.398594	17.16	19.44

3.912	1.178	0.02306	0.01116	0.5368	0.3812	9.581814	7.266342	18.82	30.64
3.104	0.9054	0.02701	0.01757	0.6461	0.4541	11.73267	8.771938	18.53	21.71
4.513	1.321	0.0202	0.009188	0.468	0.341	4.961044	5.324801	20.77	34.52
4.439	1.402	0.01912	0.009446	0.4395	0.3241	15.11906	7.600254	16.25	30.65
2.649	1.035	0.02335	0.01459	0.5522	0.3499	35.06407	3.833046	10.3	22.12
3.208	1.03	0.02374	0.01386	0.5648	0.397	17.78109	6.66961	16	25.44
2.787	0.8736	0.02561	0.01833	0.6245	0.4228	19.66074	7.102511	15.86	19.7
2.827	0.8713	0.02591	0.01839	0.5968	0.3924	18.78814	11.35961	14.9	16.16
3.238	0.877	0.0281	0.01941	0.6039	0.4111	12.89696	11.16532	15.69	16.09
3.536	1.048	0.02522	0.01492	0.5724	0.4032	12.22614	10.95385	17.12	21.91
2.769	0.944	0.02447	0.01577	0.5674	0.3637	26.50917	6.05804	12.41	20.17
2.868	1.027	0.02309	0.01514	0.5626	0.3816	26.4977	6.355954	13.47	22.17
3.609	1.22	0.0223	0.01164	0.5137	0.3619	24.19396	6.028198	13.02	28.17
2.942	1.058	0.02278	0.01349	0.5479	0.3733	25.80318	6.217819	13.41	24.69
2.957	0.9055	0.02624	0.0165	0.5934	0.382	19.37643	7.198178	14.29	19.74
3.8	0.9907	0.02906	0.01739	0.6345	0.4555	7.873889	7.776982	18.14	22.5
3.864	1.145	0.02331	0.01132	0.5241	0.3816	11.90455	5.565416	17.05	31
3.225	0.8679	0.02884	0.01792	0.61	0.3986	14.02152	7.687068	14.91	18.62

ANEXO II

Código computacional do programa controle e monitoração implementada em C++ Builder 4.0 da Borland.

```
//-----  
---#include <vcl.h>  
#pragma hdrstop  
#include "UMain.h"  
#include "math.h"  
#include "dos.h"  
#include "Fis.h"  
#include "URead.h"  
#include <stdio.h>  
#include "UBprop.h"  
//-----  
---  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
//-----  
---  
static double *pr;  
static double *hl;  
static double *hv;  
static int cont=0, i=0,j=0;  
//-----  
---  
#define LPT1 0x378  
#define Max 2  
  
unsigned int Time = 0; // Tempo atual do controle em Milesegundos  
typedef void (* DLL_EP)(void);  
#define DLL_EP_PTR __stdcall*  
  
DLL_EP Entalpia, Entropia, Pressao, Temperatura, t_sat;  
  
HINSTANCE hinstLib, _hinstLib;  
  
//----- Variáveis Globais -----  
---  
void tiempo();  
//----- Variáveis Globais -----  
---  
struct time tempo;  
char *fisname, *fisname1;  
FIS *fis;  
FIS *fis1;  
Eev *ERead;  
FILE *dados, *fProp;  
FILE *Ciclo;  
static double *e_cdt, *e_cdp, *s_cdt, *s_cdp, *e_evt, *e_evp, *s_evt, *s_evp;  
static double *te_ev, *ts_ev, *te_cd, *ts_cd;  
float he_cd, p_sat, hs_cd, he_ev, hs_ev, t_ev, t_cd, t_sp, t_sb;  
float t=0, p=0, x=0, te, ts, pe, ps, APPR_ev, APPR_cd, DTw_e, DTw_c;  
int flag =0;  
float ma_cd, ma_ev, mf_cd, mf_ev;  
int erro_ev=0, erro_cd=0, erro_cp=0, erro_val=0;
```

```

//-----
---
void GetCiclo();
double * fvetor(int n);
TFHVAC *FHVAC;

matriz *mref_tev, *UAe_APPRe, *UAe_mref;

//-----
--- __fastcall TFHVAC::TFHVAC(TComponent* Owner)
    : TForm(Owner)
{
    GetCiclo();
    for(int i = 0; i <= 999; i++) {
        Series19->AddXY(hl[i],pr[i], "",clRed);
        Series19->AddXY(hv[i],pr[i], "",clRed);
    }
    e_cdt = fvetor(Max);
    e_cdp = fvetor(Max);
    s_cdt = fvetor(Max);
    s_cdp = fvetor(Max);
    e_evt = fvetor(Max);
    e_evp = fvetor(Max);
    s_evt = fvetor(Max);
    s_evp = fvetor(Max);
    te_ev = fvetor(Max);
    ts_ev = fvetor(Max);
    te_cd = fvetor(Max);
    ts_cd = fvetor(Max);
    delete[] pr, hl, hv;
    char *dll_name = "fluid.dll";
    char * dll_sat = "T_sat.dll";
    hinstLib = LoadLibrary(dll_name);
    if(hinstLib != NULL) {
        Entalpia = (DLL_EP) GetProcAddress(hinstLib, "enthalpy");
        Entropia = (DLL_EP) GetProcAddress(hinstLib, "entropy");
        Pressao = (DLL_EP) GetProcAddress(hinstLib, "pressure");
        Temperatura=(DLL_EP) GetProcAddress(hinstLib, "temperature");
    }
    _hinstLib = LoadLibrary(dll_sat);
    if(_hinstLib != NULL)
        t_sat = (DLL_EP) GetProcAddress(_hinstLib, "t_sat");

    mref_tev = create_matriz(2, 49);
    UAe_APPRe= create_matriz(2, 49);
    UAe_mref= create_matriz(2, 49);
}
//-----
double * fvetor(int n)
{
    return new double[n];
}
//-----
---
void GetCiclo()
{
    double P, Hl, Hv;
    if((Ciclo = fopen("r134a.prn", "r"))== NULL)
        ShowMessage("Arquivo de ciclos não existente! ");
}

```

```

else {
    pr = fvetor(1000);
    hl= fvetor(1000);
    hv= fvetor(1000);
    for(int i=0; i<=999; i++) {
        fscanf(Ciclo,"%lf %lf %lf",&P, &Hl, &Hv);
        pr[i] = P;
        hl[i] = Hl;
        hv[i] = Hv;
    }
    fclose(Ciclo);
}
}
//-----
----
//Captura o tempo do sistema
double temporizador()
{
    gettimeofday(&tempo);
    static int hh = tempo.ti_hour;
    static int mm = tempo.ti_min;
    static int ss = tempo.ti_sec;
    long int h = tempo.ti_hour;
    long int m = tempo.ti_min;
    long int s = tempo.ti_sec;
    double ttt = ((h-hh)*3600 + (m-mm)*60) + (s-ss);
    return ttt;
}
//-----
----float fmedia(double *x)
{
    float soma=0;
    for(int i =0; i < Max; i++)
        soma += x[i];
    return soma/Max;
}
//-----
----
// Função que calcula o vazão do refrigerante (utilizando o balanço de
energia)
// 1 lei
float Balanco(float Te, float Ts, float he, float hs, float ma)
{
    float DeltaH = 4181 * abs(Te - Ts);
    float mf = (ma * DeltaH)/abs((hs-he)*1000);
    return mf;
}
//-----
----double COP (double h1, double h2, double h4)
{
    return ((h1-h4)/(h2-h1));
}
//-----
----double Nisentropic (double t, double p_su, double p_ex, double h1, double
h2)
{
    double s = ((float (DLL_EP_PTR)(float, float, int, int, int&)) Entropia)
(t, p/1e3, 134, 12, flag);
    double h2_s = ((float (DLL_EP_PTR)(float, float, int, int, int&))
Entalpia) (p/1e3, s, 134, 24, flag);
}

```

```

    return ((h2_s-h1)/(h2-h1));
}
//-----
---
double Q(double mref, double Deltah)
{
    return mref* abs(Deltah);
}
//-----
---
double LMTD_ev(double T_ev, double T_su_w, double T_ex_w)
{
    return ((T_su_w-T_ev)- (T_ex_w-T_ev))/log((T_su_w-T_ev)/(T_ex_w-T_ev));
}
//-----
---
double LMTD_cd(double T_cd, double T_su_w, double T_ex_w)
{
    return ((T_cd-T_su_w)- (T_cd-T_ex_w))/log((T_cd-T_su_w)/(T_cd-T_ex_w));
}
//-----
---
double n_motor (double COP,double Q_ev, double Q_cd, double)
{
    double power = 1000*(Q_cd - Q_ev);
    return (COP/(Q_ev/power));
}
//-----
---
void TFHVAC::Salvar(vetor v)
{
    static double Cop, UA_cd, UA_ev, Q_ev, Q_cd, LMTD_e, LMTD_c, N_isentropic;
    // double tt = temporizador();
    if (CBSalvar->Checked == true)
    {
        for(int i=0; i <= 14; i++)
            fprintf(dados, "%lf ", v[i]);
        // fprintf(dados, "%lf\n", tt);
    }
    if (cont <= Max)
    {
        e_cdp[cont] = v[14];
        e_cdt[cont] = v[7];
        s_cdp[cont] = v[12];
        s_cdt[cont] = v[5];
        e_evp[cont] = v[8];
        e_evt[cont] = v[0];
        s_evp[cont] = v[9];
        s_evt[cont] = v[2];

        te_cd[cont] = v[4];
        ts_cd[cont] = v[6];
        te_ev[cont] = v[3];
        ts_ev[cont] = v[1];
        cont++;
    }
    if (cont >= Max) {
        // Calculo
        t = fmedia(e_cdt);
        p = ps = fmedia(e_cdp);
    }
}

```

```

    he_cd = ((float(DLL_EP_PTR)(float, float, int, int, int&))Entalpia)
p/1e3, 134, 12, flag);
    Series20->AddXY(he_cd,p,"",clYellow);
    Series21->AddXY(he_cd,p,"",clYellow);
    t = fmedia(s_cdt);
    p = fmedia(s_cdp);
    t_cd = ((float(DLL_EP_PTR)(float, float, int, int, int&
Temperatura)(p/1e3, x, 134, 25, flag);
    p_sat = ((float(DLL_EP_PTR)(float, float, int, int, int&
Pressao)(t, x, 134, 15, flag);
    hs_cd = ((float(DLL_EP_PTR)(float, float, int, int, int& )) Entalpia
(t, x, 134, 15, flag);
    Series20->AddXY(hs_cd,p,"",clYellow);
    Series21->AddXY(hs_cd,p,"",clYellow);
    t = fmedia(e_evt);
    p = fmedia(e_evp);
    he_ev = hs_cd;
    Series20->AddXY(he_ev,p,"",clYellow);
    t = fmedia(s_evt);
    pe=p= fmedia(s_evp);
    hs_ev = ((float(DLL_EP_PTR)(float, float, int, int, int& )) Entalpia
(t, p/1e3, 134, 12, flag);
    Series20->AddXY(hs_ev,p,"",clYellow);
    t_ev =((float(DLL_EP_PTR)(float, float, int, int, int& )) Temperatura)
(p/1e3, x, 134, 25, flag); // temperatura de evaporacao
    ma_cd = EF_cd->Text.ToDouble()/ETVazao->Text.ToDouble();
    ma_ev = EF_ev->Text.ToDouble()/ETVazao->Text.ToDouble();
    Series22->AddY(t_ev,"",clBlue); // Grafico da temperatura de evaporação
    Series16->AddY(t_cd,"",clRed); // Grafico da temperatura de
condensação
    te = fmedia(te_cd);
    ts = fmedia(ts_cd);
    DTw_c = ts-te;
    Series18->AddY(DTw_c,"",clRed); // Gráfico da diferença de temperatura
da água no condensador
    APPR_cd = t_cd-ts;
    Series23->AddY(APPR_cd,"",clRed); //Gráfico da diferença (T_cd-
T_w_ex_cd)
    te = fmedia(te_ev);
    ts = fmedia(ts_ev);
    DTw_e = te-ts;
    Series17->AddY(DTw_e,"",clBlue); // Gráfico da diferença temperatura da
água no evaporação
    APPR_ev = ts-t_ev;
    Series24->AddY(APPR_ev,"",clBlue); //Gráfico da diferença (T_w_ex_ev-
T_ev)
    if (ChFalha->Checked == true) {
        /* Caculo do processamento Termico */
        Cop = COP(hs_ev,he_cd,he_ev);
        N_isentropic = Nisentropic (t,pe, ps, hs_ev, he_cd);
        te = fmedia(te_cd);
        ts = fmedia(ts_cd);
        LMTD_c = LMTD_cd(t_cd, te, ts);
        t_sb = t_cd - ts; //Grau de sub resfriamento
        mf_cd = Balanco(te, ts,he_cd, hs_cd, ma_cd);
        te = fmedia(te_ev);
        ts = fmedia(ts_ev);
        LMTD_e = LMTD_ev(t_ev, te, ts);
        t_sp = ts -t_ev; //Grau de super aquecimento
        mf_ev = Balanco(te, ts,he_ev, hs_ev,ma_ev);

```

```

Q_cd = Q(mf_cd, (hs_cd-he_cd));
Q_ev = Q(mf_ev, (hs_ev-he_ev));
UA_cd = Q_ev/LMTD_e;
UA_ev = Q_cd/LMTD_c;
if(j<=49){
    i=0;
    //X
    mref_tev->Matriz[i][j] = abs(mf_cd - mf_ev)/2;
    UAe_APPRe->Matriz[i][j] = UA_ev;
    UAe_mref->Matriz[i][j] = UA_ev;
    //Y
    mref_tev->Matriz[i+1][j] = t_ev;
    UAe_APPRe->Matriz[i][j] = APPR_ev;
    UAe_mref->Matriz[i][j+1] = abs(mf_cd - mf_ev)/2;
}
else{
    if()

}
}
}
else {
    te = fmedia(te_cd);
    ts = fmedia(ts_cd);
    t_sb = t_cd - ts;
    mf_cd = Balanco(te, ts, he_cd, hs_cd, ma_cd); //Grau de sub resfriamento
    te = fmedia(te_ev);
    ts = fmedia(ts_ev);
    t_sp = ts - t_ev;
    mf_ev = Balanco(te, ts, he_ev, hs_ev, ma_ev); //Grau de super aquecimento
}
EError->Text =FloatToStr((1-(mf_ev/mf_cd))*100);
EFR_cd->Text = FloatToStr(mf_cd);
EFR_ev->Text = FloatToStr(mf_ev);
Ltev->Caption = FloatToStr(t_ev);
Ltc_d->Caption = FloatToStr(t_cd);
cont = 0;
}
}
}

//-----
void __fastcall TFHVAC::TimerGrafTimer(TObject *Sender)
{
    /* Mostra os graficos em tempo real */
    ERead->Executar();
    double tt = temporizador();
    // Graficos das variaves de controle
    Series9->AddXY(tt, StrToFloat(ECompe->Text), "", clRed);
    Series10->AddXY(tt, StrToFloat(ETsat->Text), "", clBlue);
    Series11->AddXY(tt, StrToFloat(ECont->Text), "", clYellow);

    // Graficos de controle de Pressão
    Series12->AddXY(tt, StrToFloat(EPcomp_E->Text), "", clRed);
    Series13->AddXY(tt, StrToFloat(EPcomp_S->Text), "", clGreen);
    Series14->AddXY(tt, StrToFloat(EPval_E->Text), "", clYellow);
    Series15->AddXY(tt, StrToFloat(EPval_S->Text), "", clBlue);

    // Graficos de Temperaturas do Evaporador
    Series1->AddXY(tt, StrToFloat(EValS->Text), "", clRed);
    Series2->AddXY(tt, StrToFloat(ECompe->Text), "", clGreen);
}

```



```

Series3->AddXY(tt, StrToFloat (EEvap_AE->Text), "", clYellow);
Series4->AddXY(tt, StrToFloat (EEvap_AS->Text), "", clBlue);
// Graficos de Temperaturas condensador
Series5->AddXY(tt, StrToFloat (ECompS->Text), "", clRed);
Series6->AddXY(tt, StrToFloat (EValE->Text), "", clGreen);
Series7->AddXY(tt, StrToFloat (EEvap_AE->Text), "", clYellow);
// Series7->AddXY(tt, StrToFloat (ECond_AE->Text), "", clYellow);
Series8->AddXY(tt, StrToFloat (ECond_AS->Text), "", clBlue);
// Graficos de comportamento
// Series16->AddXY(tt, StrToFloat (ECond_AS->Text), "", clBlue);
// Series17->AddXY(tt, StrToFloat (EEvap_AS->Text), "", clBlue);
// Series18->AddXY(tt, StrToFloat (ECompS->Text), "", clBlue);
}
//-----
void __fastcall TFHVAC::Iniciar1Click(TObject *Sender)
{
TimerGraf->Enabled = true;
dados = fopen("Dados.dat", "w+");
fProp = fopen("FProp.dat", "w+");
fprintf(dados, "%s          %s          %s          %s          %s"
, "Tout_valvula", "Tout_A_evap", "Tin_comp", "Tin_A_evap", "T_A_Ambiente");
fprintf(dados, "          %s          %s          %s          %s"
%s, "Tin_valvula", "Tout_A_cond", "Tin_cond", "Pout_valvula", "Pin_comp");
fprintf(dados, "          %s          %s          %s          %s"
%s, "T_Sat", "Cop", "Pin_valvula", "Controle", "Pout_comp", "Tempo\n");
fprintf(fProp, "%s %s %s %s %s %s %s %s %s %s %s %s %s %s %s %s"
, "T_ev", "T_cd", "he_cd", "hs_cd", "mf_cd", "ma_cd", "he_ev", "hs_ev", "mf_ev", "ma_ev"
, "t_sp", "t_sb\n");
}
//-----

void __fastcall TFHVAC::Parar1Click(TObject *Sender)
{
TimerGraf->Enabled = false;
fclose(dados);
fclose(fProp);
}
//-----

void __fastcall TFHVAC::ExitClick(TObject *Sender)
{
Close();
}
//-----

void __fastcall TFHVAC::FormClose(TObject *Sender, TCloseAction &Action)
{
fclose(dados);
fclose(fProp);
FreeLibrary(hinstLib);
delete[] e_cdt, e_cdp, s_cdt, s_cdp, e_evt, e_evp, s_evt, s_evp;
delete[] te_ev, ts_ev, te_cd, ts_cd;
}
//-----
void __fastcall TFHVAC::SBControleClick(TObject *Sender)
{
SavePicture->Execute();
if(SavePicture->FileName != "\0")
ChControle->SaveToBitmapFile(SavePicture->FileName);
}

```

```

}
//-----

void __fastcall TFHVAC::SBPressaoClick(TObject *Sender)
{
    SavePicture->Execute();
    if(SavePicture->FileName != "\\0")
        ChPressao->SaveToBitmapFile(SavePicture->FileName);
}
//-----

void __fastcall TFHVAC::SBEvapClick(TObject *Sender)
{
    SavePicture->Execute();
    if(SavePicture->FileName != "\\0")
        ChEvap->SaveToBitmapFile(SavePicture->FileName);
}
//-----

void __fastcall TFHVAC::SBCondClick(TObject *Sender)
{
    SavePicture->Execute();
    if(SavePicture->FileName != "\\0")
        ChCond->SaveToBitmapFile(SavePicture->FileName);
}
//-----

void __fastcall TFHVAC::SBPropClick(TObject *Sender)
{
    SavePicture->Execute();
    if(SavePicture->FileName != "\\0")
        ChProp->SaveToBitmapFile(SavePicture->FileName);
}
//-----

void __fastcall TFHVAC::CBCalculoClick(TObject *Sender)
{
    cont=0;
}
//-----

void __fastcall TFHVAC::SBClearClick(TObject *Sender)
{
    Series20->Clear();
    Series21->Clear();
}
//-----

UNIT READ

//-----

//-----
#include <vcl.h>
#pragma hdrstop

#include "URead.h"
#include "UAsm.hpp"
#include "math.h"
#include "UMain.h"

```

```

#include <iostream.h>
#define PATM 92.48
#define kPa 98.07
//-----
#pragma package(smart_init)

//-----

#define dato 0x0
#define base 0x300

/*~~~~~*/
/*   baseout = Direccin de la puerta de salida digital   */
/*   parar = Valor del dato enviado al puerto             */
/*   girocw = Valor del dato enviado al puerto           */
/*   giroccw = Valor del dato enviado al puerto         */
/*******/
/*   Declaracion de variables   */
#define baseout 0x30B
#define parar 0x00
#define girocw 0x03
#define giroccw 0x01

/*-----*/
/*               declaracion de funciones                 */
/*-----*/
/*-----*/

void Evv::espera(void)      /*   Da un tiempo de espera para que el DAC*/
                           /*   termine su conversion                 */
{
    Sleep(40);
}

//-----
float Evv::lecanal(void)    /*   Hace la lectura del canal y
                           entrega el valor en voltios*/
{
    double lectura = InportB(base+3);
    double lectura2 = InportB(base+4);
    double total = lectura+256*lectura2;
    double volts = -10.0001+0.00488377*total;
    return(volts);
}

//-----
AnsiString Evv::SetNumber(double value) /*Coloca solo los tres digitos
                                        ms significativos*/
{
    return FloatToStrF(value, ffGeneral, 4,5);
}

//-----
void Evv::Executar(void)    /*Faz a leitura dos sensores e os
apresenta                  janela do monitor*/
{
    static double v[14];
    for(int w = 0; w <=13 ; w++)

```

```

{
  double temp, pres, Tsat, P;
  double temp_1, temp_2, temp_3, temp_4, temp_5, temp_6, temp_7,
temp_8,
  double pres_1, pres_2, pres_3, pres_4;
  double encod, Controle;
  double COPcar;
  OutportB(0x309, dato);
  OutportB(base, dato);
  OutportB(base+1, dato+w);
  espera();
  lecanal();
  switch (w+1)
  {
    case 1 : {
      temp = lecanal();
      temp_1=30.95362-7.421543*temp;
      v[0] = temp_1;
      FHVAC->EValS->Text = SetNumber(temp_1);
    }
    break;
    case 2 : {
      temp = lecanal();
      temp_2=30.919 - 7.2348*temp;
      v[1] = temp_2;
      FHVAC->EEvap_AS->Text = SetNumber(temp_2);
    }
    break;
    case 3 : {
      temp = lecanal();
      temp_3=29.95444-7.295898*temp; v[2] = temp_3;
      FHVAC->ECompE->Text = SetNumber(temp_3);
    }
    break;
    case 4 : {
      temp = lecanal();
      temp_4=33.57 - 7.0921*temp;
      v[3] = temp_4;
      FHVAC->EEvap_AE->Text = SetNumber(temp_4);
    }
    break;
    case 5 : {
      temp = lecanal();
      temp_5=31.48902-7.385725*temp;
      v[4] = temp_4;
      FHVAC->ECond_AE->Text = SetNumber(temp_5);
    }
    break;
    case 6 : {
      temp = lecanal();
      temp_6=31.48902-7.385725*temp; v[5] = temp_6;
      FHVAC->EValE->Text = SetNumber(temp_6);
    }
    break;
    case 7 : {
      temp = lecanal();
      temp_7=33.666 - 7.2634*temp;
      v[6] = temp_7;
      FHVAC->ECond_AS->Text = SetNumber(temp_7);
    }
  }
}

```

```

        break;
    case 8 : {
        temp = lecanal();
        temp_8=53.92891-7.215136*temp; v[7] = temp_8;
        FHVAC->ECompS->Text = SetNumber(temp_8);
    }
    break;
    case 9 : {
        pres = lecanal();
        pres_1=-2.506707 + 2.526877*pres; v[8] = pres_1*kPa +
PATM;
        FHVAC->EPval_S->Text = SetNumber(v[8]);
    }
    break;
    case 10: {
        pres = lecanal();
        pres_2=-2.473712 + 2.534405*pres; v[9] = pres_2*kPa +
PATM;
        FHVAC->EPcomp_E->Text = SetNumber(v[9]);
    }
    break;
    case 11: {
        P=(pres_1+1)*1.02;
        Tsat=(-5.8961E+01+46.94001*P-1.8841E+01*pow(P,2)+
5.098956*pow(P,3)-7.3909E-01*pow(P,4)+0.04339722*pow(P,5))+7;
        v[10] = Tsat;
        FHVAC->ETsat->Text = SetNumber(Tsat);
        COPcar = (temp_1+273.3)/(temp_6-temp_1);
        v[11] = COPcar;
        FHVAC->ECOP->Text = SetNumber(COPcar);
    }
    break;
    case 12: {
        pres = lecanal();
        pres_4=-6.285802 + 6.300629*pres; v[12] = pres_4*kPa +
PATM;
        FHVAC->EPval_E->Text = SetNumber(v[12]);
    }
    break;
    case 13: {
        encod = lecanal();
        Controle=(0.01422383-
0.2001864*encod+1.109372*pow(encod,2)-0.2604147*pow(encod,3)-1.29);
        v[13] = Controle;
        FHVAC->ECont->Text = SetNumber(Controle);
    }
    break;
    case 14: {
        pres = lecanal();
        pres_3=-6.332863+6.2534*pres; v[14] = pres_3*kPa + PATM;
        FHVAC->EPcomp_S->Text = SetNumber(v[14]);
    }
    break;
}
}
FHVAC->Salvar(v);
}
//-----

```



```

//-----
//-----

UNIT BPROP

//-----
#include <vcl.h>
#pragma hcrstop
#include "UBprop.h"
#include "stdio.h"
#include "math.h"
#include "stdlib.h"

//-----
#pragma package(smart_init)

//-----
void GetDouble(FILE *inFile, matriz *x)
{
    char *str;
    fscanf(inFile,"%s",&str);
    for(int i=0;i < x->Cols; i++)
        for(int j=0;j < x->Rows ; j++)
            fscanf(inFile,"%lf", &x->Matriz[j][i]);
}

//-----
double Logistico(double x)
{
    return(1.0 / (1.0 + exp(-x)));
}

//-----
void GetScale(matriz *x, matriz *y, matriz *xm, matriz * xb)
{ // Escalonando entre 0.1 e 0.9 na equação linear  $y = m*x + b$ 
  /* 0.9 = max(x) + b
     0.1 = min(x) + b */

    matriz * max = create_matriz(x->Rows, 1);
    matriz * min = create_matriz(x->Rows, 1);
    max = GetMax(x);
    min = GetMin(x);
    for(int i=0; i < max->Rows; i++)
        for(int j=0; j < max->Cols; j++) {
            xm->Matriz[i][j] = 0.9/(max->Matriz[i][j]-min->Matriz[i][j]);
            xb->Matriz[i][j] = 0.1 - xm->Matriz[i][j]* min->Matriz[i][j];
        }
    for(int i=0; i < y->Rows; i++)
        for(int j=0; j < y->Cols; j++)
            y->Matriz[i][j] = xm->Matriz[i][0] * x->Matriz[i][j] + xb-
>Matriz[i][0];
}

//-----
double Out_Scale(double y, double m, double b)
{

```

```

    //  $y = m \cdot x + b \rightarrow x = (y - b) / m$ 
    return (y-b) / m;
}

//-----
matriz * GetMax(matriz * x)
{
    matriz * max = create_matriz(x->Rows, 1);
    double aux=0;
    for(int i=0; i<x->Rows; i++) {
        aux = x->Matriz[i][0];
        for(int j=0; j<x->Cols; j++)
            if (aux < x->Matriz[i][j]) aux = x->Matriz[i][j];
        max->Matriz[i][0] = aux;
    }
    return max;
}

//-----
matriz * GetMin(matriz * x)
{
    matriz * min = create_matriz(x->Rows, 1);
    double aux=0;
    for(int i=0; i<x->Rows; i++) {
        aux = x->Matriz[i][0];
        for(int j=0; j<x->Cols; j++)
            if (aux > x->Matriz[i][j]) aux = x->Matriz[i][j];
        min->Matriz[i][0] = aux;
    }
    return min;
}

//-----
matriz *create_matriz(int Rows,int Cols)
{
    matriz *y;
    int i;
    y = new matriz;
    y->Matriz = new double *[Rows];
    y->Rows = Rows;
    y->Cols = Cols;
    for(i=0; i<Rows; i++)
        y->Matriz[i] = new double [Cols];

    return(y);
}

//-----
void matriz_multiply(matriz *Left,matriz *Right,matriz *Product)
{
    int i,j,k;
    double sum;
    for(i=0; i<Left->Rows; i++)
    {
        for(j=0; j<Right->Cols; j++)
        {
            sum=0;
            for(k=0; k<Left->Cols; k++)
                sum += Left->Matriz[i][k] * Right->Matriz[k][j];
        }
    }
}

```

```

        Product->Matriz[i][j] = sum;
    }
}

//-----
matriz * matriz_multiply2(matriz *Left,matriz *Right)
{
    int i,j,k;
    double sum;
    static matriz *Product;
    Product = create_matriz(Left->Rows, Right->Cols);
    for(i=0; i<Left->Rows; i++)
    {
        for(j=0; j<Right->Cols; j++)
        {
            sum=0;
            for(k=0; k<Left->Cols; k++)
                sum += Left->Matriz[i][k] * Right->Matriz[k][j];

            Product->Matriz[i][j] = sum;
        }
    }
    return Product;
}

//-----
matriz * matriz_sub(matriz *Left,matriz *Right,matriz *Product)
{
    matriz * Sub = create_matriz(Right->Rows,Right->Cols);
    for(int i=0; i<Right->Rows; i++)
        for(int j=0; j<Right->Cols; j++)
            Sub->Matriz[i][j]= Right->Matriz[i][j] - Left->Matriz[i][j];
    return Sub;
}

//-----
matriz * operador (char c, int n, matriz *Left)
{
    matriz * Sub = create_matriz(Left->Rows,Left->Cols);
    for(int i=0; i< Left->Rows; i++)
        for(int j=0; j< Left->Cols; j++) {
            switch(c) {
                case '*' :    Sub->Matriz[i][j]= n * Left->Matriz[i][j];
break;
                case '/' :    Sub->Matriz[i][j]= n / Left->Matriz[i][j];
break;
                case '+' :    Sub->Matriz[i][j]= n + Left->Matriz[i][j];
break;
                case '-' :    Sub->Matriz[i][j]= n - Left->Matriz[i][j];
break;
                case '^' :    Sub->Matriz[i][j]= pow(Left->Matriz[i][j],n);
break;
                default: break;
            }
        }
    return Sub;
}
//-----

```

```

matriz * Ones(matriz *Right)
{
    int i,j;
    matriz *Left = create_matriz(Right->Rows+1, Right->Cols);
    for(i=0; i<Right->Rows +1; i++)
        for(j=0; j<Right->Cols; j++)
            if(i==0) Left->Matriz[i][j] = 1;
            else Left->Matriz[i][j] = Right->Matriz[i-1][j];
    return Left;
}
//-----
matriz * GetLogistic(matriz *Right)
{
    matriz *Logistic = create_matriz(Right->Rows, Right->Cols);
    for(int i=0; i<Right->Rows; i++)
        for(int j=0; j<Right->Cols; j++)
            Logistic->Matriz[i][j] = Logistico(Right->Matriz[i][j]);
    return Logistic;
}
//-----
matriz * GetOut_Scale(matriz *Right, double tm, double ts)
{
    matriz *Scale = create_matriz(Right->Rows, Right->Cols);
    for(int i=0; i<Right->Rows; i++)
        for(int j=0; j<Right->Cols; j++)
            Scale->Matriz[i][j] = Out_Scale(Right->Matriz[i][j], tm, ts);
    return Scale;
}
//-----

void GetDados(char * FileName, matriz * x, matriz* W1,matriz *W2, double xm,
double xb)
{
    FILE * inFile;
    int N, in, med, out;
    if((inFile = fopen(FileName, "r"))== NULL) {
        ShowMessage("Arquivo não encontrado");
        exit(1);
    }
    fscanf(inFile,"%d",&N);
    fscanf(inFile,"%d",&in);
    fscanf(inFile,"%d",&med);
    fscanf(inFile,"%d",&out);
    GetDouble(inFile,x = create_matriz(in, N));
    GetDouble(inFile,W1 = create_matriz(med, in+1));
    GetDouble(inFile,W2 = create_matriz(out, med+1));
    fscanf(inFile,"%lf %lf", xm, xb);
}
//-----

```