

TESE

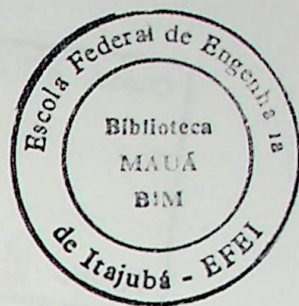
999

**Escola Federal de Engenharia  
de Itajubá**

**Supervisão e Proteção de  
um Sistema UPS**

**Itajubá - MG**

**Dezembro 1998**



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

# Supervisão e Proteção de um Sistema UPS

## DISSERTAÇÃO DE MESTRADO

**AUTOR:** *Luciano Machado Cavalca*

**ORIENTADOR:** *Prof. Luiz Eduardo Borges da Silva*

**CO-ORIENTADOR:** *Prof. Valberto Ferreira da Silva*

Trabalho de Dissertação apresentado à comissão de pós-graduação da Escola Federal de Engenharia de Itajubá como parte dos requisitos necessários à obtenção do título de Mestre em Ciências em Engenharia Elétrica.

Itajubá, Novembro de 1998

Crer é ver além das aparências,  
procurar e querer ver o infinito.

Ter confiança é sentirmos Deus perto  
de nós ainda que o céu esteja escuro  
e o horizonte distante.

A confiança é companheira da  
fé, amiga nas situações difíceis.

Não temas, porque estou contigo.

Pequenas renúncias ajudam no  
crescimento da personalidade.

Deus não olha para o valor, mas para  
o amor com que faço alguma coisa.

## AGRADECIMENTOS

A DEUS: que sustentou os meus passos cansados, os meus momentos de desânimo, que ergueu-me dos obstáculos. DEDICO A TI, SENHOR, ESTA ETAPA VENCIDA.

A minha esposa MARIA DAS GRAÇAS K. S. CAVALCA, pelo amor, dedicação e sacrifício oferecidos a mim, durante esta página de nossas vidas.

A minha filha Suzane que esteve aqui no início do mestrado e agora do lado de Jesus , tenho certeza continuou rezando por mim. E também aos meus filhos Tiago e Mariele que com sorrisos e carinho me ajudaram nesta batalha.

Aos meus pais GENIVALDO CAVALCA e EDNA MACHADO CAVALCA, por propiciar orientação familiar e educacional exemplar ao longo de minha vida.

Ao Prof. LUIZ EDUARDO B. DA SILVA, por consentir na orientação, solícito e competente nos momentos em que foi por mim procurado.

Ao Prof. VALBERTO FERREIRA DA SILVA, primando em todos os momentos por competência, interesse e muita amizade.

Ao Prof. LUIZ LENARTH GABRIEL VERMAAS. inestimável colaborador e grande amigo em todas as fases de elaboração deste trabalho.

Aos mestrandos MÍLTON EVANGELISTA DE OLIVEIRA FILHO e PAULO ROBERTO SANTIAGO, pela contribuição decisiva na fase de elaboração da etapa computacional da tese.

A todos os funcionários do Departamento de Eletrônica da EFEI, pela atenção, amizade e ajuda que a mim dispensaram, ao longo de todo período acadêmico.

À CAPES, pelo financiamento deste trabalho.

Em suma, a todos que direta ou indiretamente colaboraram para este sucesso.



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## RESUMO

## RESUMO

Os sistemas denominados de fornecimento contínuo de potência, cuja sigla adotada é UPS, são fundamentais à alimentação de alguns tipos de cargas. como, Centro de Processamento de Dados, seja industrial ou bancário, fábricas(papel, vidro, etc), são exemplos típicos de instalação que necessitam de fornecimento contínuo de energia elétrica.

O presente trabalho descreve um software para detecção e conseqüente sinalização de modo a se obter a atuação de um sistema de proteção, face à ocorrência de uma ou mais falhas em quaisquer das unidades de potência de um Sistema de Fornecimento Contínuo de Potência(UPS).

É feita uma apresentação das filosofias de estudo e atitudes de atuação perante falhas existentes nos circuitos de potência típicos de um UPS.

A filosofia de operação de um UPS, assim como as falhas específicas de cada unidade correspondente aos blocos também são descritas.

Enfatizando os trabalhos de laboratório e simulação desenvolvidos, são mostrados os principais tipos de circuitos eletrônicos utilizados para de se detectar os diferentes tipos de faltas. tanto na unidade retificadora quanto da unidade inversora, tipicamente existentes numa UPS.

A parte principal da tese é realmente a elaboração de um programa em linguagem C, para supervisão e proteção das falhas; este programa permite desde o envio de sinais para os circuitos de proteção, até o detalhamento da falha que é feito de forma visual. Apesar do caráter didático existente neste programa, o mesmo pode ser aproveitado em inúmeras outras aplicações. inclusive fora da área industrial propriamente dita.

Cita-se como exemplo, o interesse da MRNC(Mineradora Rio do Norte) da cidade de Porto Trombetas/Pará, em aplicá-lo em seus Inversores, os quais alimentam motores de indução trifásicos de elevada potência e, onde ocorrem constantes queimas de fusíveis e SCRs. Estas falhas na prática representam um elevado tempo de identificação dos elementos danificados e, conseqüente tempo de parada elevado dos equipamentos, acarretando um sério prejuízo financeiro.

O programa também apresenta uma série de telas, que permite ao usuário tanto definir os valores típicos de atuação da proteção, quanto a visualização e detalhamento da(s) falha(s) ocorrida(s).

Finalmente, foi feito um estudo de tempos decorridos para execução da parte de supervisão pelo programa. Estas medições foram feitas em três configurações diferentes de microcomputadores padrão IBM-PC. Os resultados das medidas de tempos foram da ordem de milissegundos, o que na prática é o que se consegue em estratégias de supervisão e proteção de equipamentos industriais.

## ABSTRACT

The denominated systems of continuous supply of power, whose adopted acronym is UPS, they are fundamental to the feeding of some types of loads. as, Center of Processing of Data, be industrial or bank, factories(paper, glass, etc), they are typical examples of installation that need continuous supply of electric energy.

The present work describes a software for detection and consequent way signalization obtaining the performance of a protection system, face to the occurrence of an or more flaws in any of the units of potency of a System of Power Continuous Supply (UPS).

It is made a presentation of the study philosophies and attitudes of performance before existent flaws in the typical potency circuits of a UPS.

The philosophy of operation of a UPS, as well as the specific flaws of each unit corresponding to the blocks is also described.

Emphasizing the laboratory works and developed simulation, the main types of electronic circuits used are shown for of detecting the different types of lacks, so much in the rectifier unit as of the inverter. unit typically existent in a UPS.

The main part of the thesis is really the elaboration of a program in language C, for supervision and protection of the flaws; this program allows from the shipping of signs for the protection circuits, until the detailment of the flaw that is done in a visual way. In spite of the existent didactic character in this program, the same can be taken advantage of in countless other applications, besides out of the industrial area properly dictates.

He/she makes an appointment as example, the interest of MRNC(Miner Rio of the North) of Porto Trombetas/Pará's city, in applying it in its Investors, which feed motors of trifasic induction high potency and, where they happen constants you burn of fuses and SCRs. These flaws in the practice represent a high time of identification of the damaged elements and, consequent elevated time of stop of the equipments, carting a serious financial damage.

The program also presents a series of screens, that allows to the so much user to define the typical values of performance of the protection, as the visualization and detailment of fail(s) occurred.

Finally, it was made a study of times elapsed for execution on behalf of supervision for the program. These mensurations were made in three configurations different from microcomputers pattern IBM-PC. The results of the measures of times went of to miliseconds order, which is in the practice what is gotten in supervision strategies and protection of industrial equipments.



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## SUMÁRIO

### 1.2 Controles da Evolução de Falha em Conversores

1.2.1 Controle da Evolução de Falha de Comutação no Inversor

1.2.2 Controle da Evolução de Falha de Comutação no Retificador

### 1.3 Estratégias de Controle

### 1.4 Falhas em Inversores do Tipo T<sub>1</sub>o

### 2 DESCRIÇÃO DE UM UPS

2.1 Definição

2.2 Classificação

2.3 Modos de Operação

2.3.1 Modo de Operação Normal

# SUMÁRIO

Resumo.....	i
Abstract.....	ii
<b>Capítulo 1</b>	<b>SUPERVISÃO E PROTEÇÃO DE CIRCUITOS COM SEMICONDUTORES DE POTÊNCIA</b>
1.1	Falhas em Pontes Tiristorizadas.....3
1.1.1	Falha de Comutação.....3
1.1.2	Disparo Aleatório por Tensão.....5
1.1.3	Condução Reversa(“backfire”).....6
1.1.4	Períodos de Condução de Semicond. para Falhas de Comutação.....6
1.1.1.4.1	Avanço do Ângulo de Fase.....6
1.1.1.4.2	Atraso do Ângulo de Fase.....6
1.1.1.4.3	Ausência de Sinal no Gate.....7
1.1.1.4.4	Ausência de Fase.....8
1.2	Controle da Evolução de Falha em Conversores.....9
1.2.1	Controle da Evolução de Falha de Comutação no Inversor.....10
1.2.2	Controle da Evolução de Falha de Comutação no Retificador.....11
1.3	Estratégias de Controle.....12
1.4	Falhas em Inversores do Tipo Tensão Imposta.....19
<b>Capítulo 2</b>	<b>DESCRIÇÃO DE UM UPS</b>
2.1	Definição.....23
2.2	Classificação.....24
2.3	Modos de Operação.....25
2.3.1	Modo de Operação Normal.....25

	2.3.2 Modo de Operação Back – Up.....	26
	2.3.3 Modo de Operação By–Pass.....	27
	2.4 Indicações Visuais de um UPS.....	30
	2.5 Configuração do Circuito de Controle de uma UPS.....	31
	2.6 Falhas Específicas dos Blocos Retificador e Inversor de um UPS.....	32
	2.6.1 Falhas Específicas do Bloco Retificador.....	32
	2.6.2 Falhas Específicas do Bloco Inversor.....	33
<b>Capítulo 3</b>	<b>HARDWARE PARA DETECÇÃO DE DEFEITOS E SUPERVISÃO</b>	
	3.1 Considerações Iniciais.....	39
	3.2 Hardware para Falhas do Sistema.....	39
	3.3 Hardware para Pontes Trifásicas Tiristorizadas.....	48
	3.4 Hardware para Inversores Tipo Fonte de Tensão.....	50
	3.5 Resultados de Simulações.....	55
<b>Capítulo 4</b>	<b>SOFTWARE DE SUPERVISÃO E PROTEÇÃO</b>	
	4.1 Considerações Iniciais.....	65
	4.2 Descrição e Caracterização de Softwares.....	65
	4.3 Perfil da Linguagem C.....	66
	4.4 Detalhamento do Programa.....	67
	4.5 Estrutura de Arquivos do Programa.....	75
<b>Capítulo 5</b>	<b>SIMULAÇÕES</b>	
	5.1 Considerações Iniciais.....	79
	5.2 Estratégia de Simulação.....	79
	5.3 Tempos de Execução do Programa.....	81

Capítulo 6	
CONCLUSÕES.....	88
Apêndice A	PROGRAMA-FONTE EM
LINGUAGEM C.....	91
Apêndice B	ARQUIVOS DE DADOS
UTILIZADOS NA SIMULAÇÃO.....	118
Apêndice C	TEMPOS DE EXECUÇÃO
DOS BLOCOS DA UPS.....	122
Apêndice D	EXEMPLO DE TELAS DE
VISUALIZAÇÃO DE FALHAS.....	127
Apêndice E	DIAGRAMA EM BLOCOS
DE UM UPS.....	131
REFERÊNCIAS BIBLIOGRÁFICAS.....	134

## ÍNDICE DE FIGURAS



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## ÍNDICE DE FIGURAS

# ÍNDICE DE FIGURAS

<b>Figura 1</b> - Falha de Comutação.....	4
<b>Figura 2</b> - Períodos de Condução de Semicondutores para Falhas de Comutação.....	5
<b>Figura 3</b> - Avanço de $\alpha$ .....	7
<b>Figura 4</b> - Ausência de Sinal de Gate( $\alpha < 90^\circ$ ).....	9
<b>Figura 5</b> - Ausência de Fases.....	8
<b>Figura 6</b> - Evolução Controlada de Falhas de Comutação Na Ponte Operando como Inversora.....	10
<b>Figura 7</b> - Evolução Controlada de Falhas de Comutação na Ponte Operando como Retificadora.....	12
<b>Figura 8</b> - Diagrama em Blocos do Sistema de Controle.....	14
<b>Figura 9</b> - Gerador de Pulsos de Disparo.....	15
<b>Figura 10</b> - Determinação da Fase dos Pulsos para o Controle Tipo IPC.....	16
<b>Figura 11</b> - Circuito do Filtro Ativo.....	17
<b>Figura 12</b> - Sistema de controle de Grade de Pulsos(PPC).....	18
<b>Figura 13</b> - Circuito de Medida do Ângulo $\alpha$ .....	19
<b>Figura 14</b> - Diagrama Em Blocos da Unidade de Disparo do GTO.....	20
<b>Figura 15</b> - Correntes e Tensões do GTO durante Condução e Bloqueio.....	21
<b>Figura 16</b> - Diagrama em Blocos do Bloco de Intetravamento Adaptativo.....	22
<b>Figura 17</b> - Diagrama em Blocos de um UPS Típico.....	24
<b>Figura 18</b> - Modos de Operação do UPS.....	25
<b>Figura 19</b> - Circuito Simplificado de um UPS.....	26
<b>Figura 20</b> - Fluxo de Energia no Modo de Operação Normal.....	27
<b>Figura 21</b> - Fluxo de Energia no Modo de Operação Back – Up.....	28
<b>Figura 22</b> - Fluxo de Energia no Modo de Operação By – Pass.....	29
<b>Figura 23</b> - Chaveamento Eletro–Mecânico.....	30
<b>Figura 24</b> - Chaveamento Estático.....	32
<b>Figura 25</b> - Configuração do Circuito de Controle.....	32

<b>Figura 26</b> - Processo de Geração de Sobrecorrente em Curto do Braço.....	36
<b>Figura 27</b> - Falta de Fase e Subtensão.....	41
<b>Figura 28</b> - Circuito para Detecção de Falta de Fase e Subtensão.....	42
<b>Figura 29</b> - Formas de Onda Típicas do Circuito da Figura 28.....	43
<b>Figura 30</b> - Circuito para Sinalização de Falta de Fase.....	44
<b>Figura 31</b> - Diagrama em Blocos de Seqüência Correta das Fases.....	45
<b>Figura 32</b> - Circuito para Detecção da Seqüência Correta de Fases.....	46
<b>Figura 33</b> - Formas de Onda na Inversão de Fase.....	47
<b>Figura 34</b> - Formas de Onda na Falta de Fase.....	47
<b>Figura 35</b> - Formas de Onda na Subtensão.....	47
<b>Figura 36</b> - Circuito Divisor Utilizado.....	48
<b>Figura 37</b> - Circuito de Detecção de Falha de Fase.....	48
<b>Figura 38</b> - Circuito de Detecção de ON/OFF e Curto – Circuito.....	50
<b>Figura 39</b> - Circuito de Detecção da Corrente de Saída DC.....	51
<b>Figura 40</b> - Circuito para Detecção de Sobrecorrente.....	52
<b>Figura 41</b> - Sinais de Entrada/Saída do Pré - Amplificador e o Comando de Bloqueio por Sobrecorrente.....	52
<b>Figura 42</b> – Circuito de Drive para IGBT.....	53
<b>Figura 43</b> – Formas de Onda do Circuito de Drive em Operação.....	55
<b>Figura 44</b> – Versão Modificada do Circuito da figura 37.....	56
<b>Figura 45</b> – Falta da Fase 3.....	58
<b>Figura 46</b> – Versão Modificada do Circuito da figura 38.....	59
<b>Figura 47</b> – SCR ON/OFF e Curto–Circuito .....	60
<b>Figura 48</b> – Versão Modificada do Circuito da figura 42.....	61
<b>Figura 49</b> – Formas de Onda para o Drive.....	62
<b>Figura 50</b> – Proteção Atuou sobre IGBT.....	64
<b>Figura 51</b> – Compartimentos do Programa.....	69
<b>Figura 52</b> – Seqüência de Análise de Parâmetros.....	70
<b>Figura 53</b> – Verificação dos Parâmetros de Start–Up.....	73
<b>Figura 54</b> – Verificação dos Parâmetros Digitais dos Blocos.....	74
<b>Figura 55</b> – Verificação dos Parâmetros Analógicos dos Blocos.....	75
<b>Figura 56</b> – Interação Entre os Arquivos.....	78
<b>Figura 57</b> – Sem Falha.....	82

<b>Figura 58</b> – Uma Falha.....	82
<b>Figura 59</b> – Todas Falhas.....	83
<b>Figura 60</b> – START – UP – Hardware 80486.....	85
<b>Figura 61</b> – START – UP – Hardware Pentium.....	85
<b>Figura 62</b> – START – UP – Hardware Pentium II.....	86
<b>Figura 63</b> – Diagrama em Blocos de um UPS.....	134

## ÍNDICE DE TABELAS



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## ÍNDICE DE TABELAS

## ÍNDICE DE TABELAS

<b>Tabela 1</b> - Causas de Sobrecorrentes e Características.....	35
<b>Tabela 2</b> - Causas de Sobrecorrente num Inversor.....	35
<b>Tabela 3</b> - Causas de Sobretensão e Características.....	37
<b>Tabela 4</b> – Comparação entre Softwares.....	67
<b>Tabela 5</b> – Parâmetros de Falha Analisados pelo Programa.....	71
<b>Tabela 6</b> – Arquivos de Dados Usados na Simulação.....	77
<b>Tabela 7</b> - Tempos Médios de Execução do Programa.....	81
<b>Tabela 8</b> –Tempos Médios de Execução dos Blocos.....	83
<b>Tabela 9</b> – Hardware 80486 Bloco Start–Up.....	122
<b>Tabela 10</b> – Hardware 80486 Bloco Inversor.....	122
<b>Tabela 11</b> – Hardware 80486 Bloco Sistema.....	123
<b>Tabela 12</b> – Hardware 80486 Bloco Retificador/Chave Estática.....	123
<b>Tabela 13</b> – Hardware Pentium Bloco Start – Up.....	124
<b>Tabela 14</b> – Hardware Pentium Bloco Inversor.....	124
<b>Tabela 15</b> – Hardware Pentium Bloco Sistema.....	125
<b>Tabela 16</b> – Hardware Pentium Bloco Retificador/Chave Estática.....	125
<b>Tabela 17</b> – Hardware Pentium II Bloco Start – Up.....	126
<b>Tabela 18</b> – Hardware Pentium II Bloco Inversor.....	126
<b>Tabela 19</b> – Hardware Pentium II Bloco Sistema.....	127
<b>Tabela 20</b> – Hardware Pentium II Bloco Retificador/Chave Estática.....	127



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## CAPÍTULO 1

# SUPERVISÃO E PROTEÇÃO DE CIRCUITOS COM SEMICONDUTORES DE POTÊNCIA

# Capítulo 1 - SUPERVISÃO E PROTEÇÃO DE CIRCUITOS COM SEMICONDUTORES DE POTÊNCIA

A rápida evolução tecnológica dos semicondutores de potência e conseqüente queda nos custos dos mesmos, levaram as empresas a utilizar tais dispositivos em equipamentos desde a área Industrial até a de Sistema de Potência, passando pela área de grandes Centros de Processamento de Dados.

Aliado a isso, sistemas muito complexos puderam ser implementados; todavia o grau de responsabilidade, em termos de segurança de operação, está levando as próprias empresas a adotarem filosofias de supervisão e proteção dos mesmos.

O presente capítulo irá classificar os tipos mais relevantes de falhas em circuitos de potência de um modo bem objetivo e rápido, para que sirva de embasamento para as análises a serem efetuadas ao longo deste trabalho.

No capítulo 2 é descrito a filosofia de operação de um UPS, bem como as falhas específicas da unidade correspondente ao bloco Inversor.

No capítulo 3, dando ênfase aos trabalhos laboratoriais e de simulação desenvolvidos, são mostrados os principais tipos de faltas, tanto na unidade retificadora quanto da unidade inversora, tipicamente encontradas num UPS.

A parte principal da tese é realmente a elaboração de um programa em linguagem C. para supervisão e proteção das falhas existentes, sendo o assunto do capítulo 4 desta dissertação. Este programa permite desde o envio de sinais para os circuitos de proteção, até o detalhamento da falha que é feito de forma visual. Apesar do caráter didático existente neste programa, o mesmo pode ser aproveitado em inúmeras outras aplicações, inclusive fora da área industrial propriamente dita.

Finalmente o capítulo 5 mostra as telas de vídeo, que o programa produz para monitoramento e aviso de falhas, além do estudo de tempos necessários para execução da parte de supervisão feita pelo programa.

## 1.1-) Falhas em Pontes Tiristorizadas

Por se tratar de uma topologia extensivamente citada na literatura, este capítulo não se preocupa em definir as topologias das unidades retificadoras e inversoras de um equipamento UPS. A unidade retificadora utiliza a tradicional ponte de Graetz, a qual é constituída de seis tiristores(SCRs) numerados de acordo com a seqüência de gatilhamento. Do mesmo modo, a unidade inversora possui a clássica topologia de um VSI(Voltage Source Inverter) ou Inversor por Tensão Imposta como é conhecida no meio técnico, a qual também é constituída por seis semicondutores de potência, somente que os mesmos são dotipo PBT(Power Bipolar Transistor), IGBT(Insulated Gate Bipolar Transistor) ou ainda o GTO(Gate Turn-Off Thyristor). A título de ilustração no apêndice E, está mostrado o diagrama em blocos detalhado de um UPS, tanto da parte de potência quanto da parte de controle.

As falhas abrangem tanto Conversores com comutação pela rede, quanto Inversores do tipo por tensão imposta. São dispositivos, controlando sinal trifásico da rede elétrica que alimenta um circuito de potência.

O primeiro estudo e, conseqüente classificação de falhas num circuito de potência, foi feito em [1]. No caso, são analisadas duas categorias básicas de falhas: o semicondutor conduzir fora de seus períodos normais de condução ou, falhar quando o mesmo deveria normalmente conduzir.

A primeira categoria inclui a condução reversa e disparo aleatório por tensão, enquanto que a outra, cobre algumas formas de falha de comutação.

### 1.1.1-) Falha de Comutação

Este tipo de falha é mostrada abaixo na figura 1. O circuito corresponde a um circuito retificador trifásico chamado ponte de Graetz, onde o SCR 5 falha ao conduzir; pode ocorrer recuperação do circuito, desde que o ângulo de disparo seja inferior a  $120^\circ$ (figura 1.b.1).

Isto é possível para operação de retificação e, a falha inicial é geralmente classificada como "Misfire". De outra forma, o semicondutor 5 continua conduzindo até sua região de operação normal(figura 1.b.2).

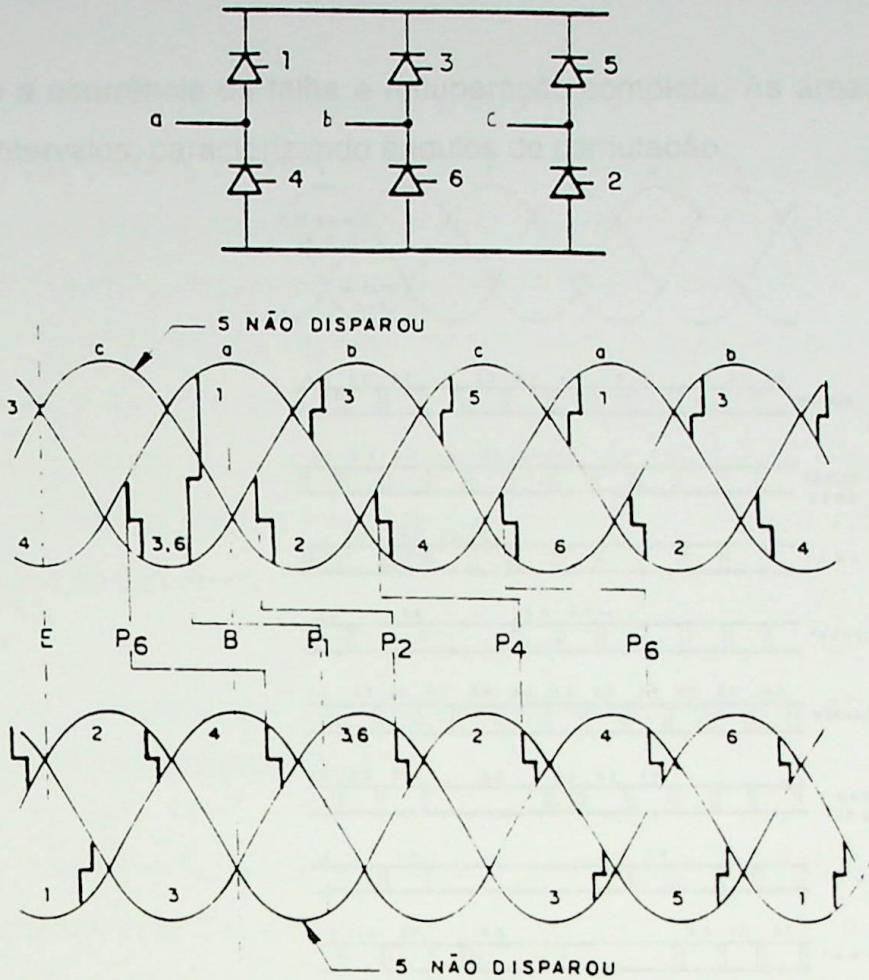


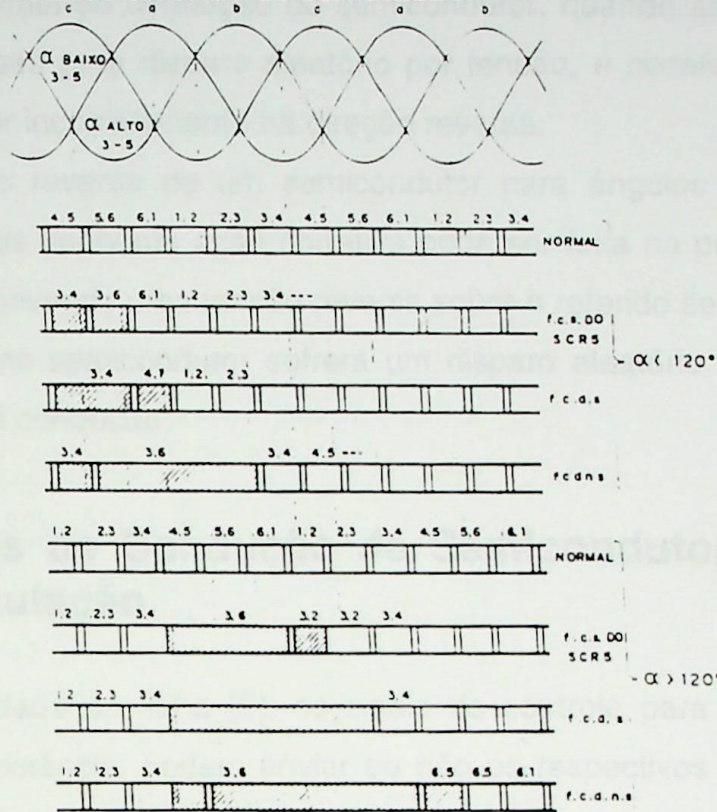
Figura 1 - Falha de Comutação  
 a) Circuito  
 b) Formas de Onda

Pode então, existir quatro possíveis conseqüências da referida falha.

A primeira é simplesmente que a falha num semicondutor poderia ser repetitiva. O segundo caso é chamado **falha de comutação duplamente sucessiva**, onde na figura 1, a falha inicial do semicondutor 5 é imediatamente seguida por uma falha do semicondutor 6 no ponto P<sub>6</sub>. O terceiro caso, chamado **falha de comutação duplamente não-sucessiva**, envolveria a primeira falha do semicondutor 5, seguida pela segunda ocorrência da falha do semicondutor 2 ao comutar corretamente com o semicondutor 6 no ponto P<sub>2</sub>. O último caso da segunda ocorrência da falha do semicondutor 2 de comutar com o semicondutor 4, ocorre quando o semicondutor 3 está neste momento conduzindo em sua região normal. Desta forma, este caso pode ser considerado como um caso separado de **falha de comutação simples**

Os períodos de condução seqüencial para este exemplo, são mostrados na figura 2. As áreas hachuradas mostram períodos de condução anormais,

entre a ocorrência da falha e recuperação completa. As áreas são demarcadas por intervalos, caracterizando ângulos de comutação.



f.c.s. – falha de comutação simples  
 f.c.s.d.s. – falha de comutação dupla sucessiva  
 f.c.d.n.s. – falha de comutação dupla não - sucessiva

Figura 2 - Períodos de Condução de Semicondutores para Falhas de Comutação

### 1.1.2-) Disparo Aleatório por Tensão

Esta falha pode ocorrer a qualquer momento, fora da região normal de condução, quando um semicondutor é submetido a uma tensão de condução. As conseqüências podem ser similares ao tipo Falha de Comutação, mas geralmente ocorrem por razões diversas.

Por definição, onde qualquer similaridade existe( ex.: por disparo espontâneo de um semicondutor logo após seu período normal de condução), um disparo aleatório por tensão de um semicondutor apareceria tal como uma falha de comutação do próximo semicondutor no mesmo ramo da ponte; isto quer dizer que falha de comutação do semicondutor 3 é equivalente ao disparo aleatório por tensão do semicondutor 1. É importante neste ponto, para efeito de classificação, distinguir entre esses tipos de falha.

### **1.1.3-) Condução Reversa(“backfire”)**

No período normal de condução do semicondutor, quando as condições de tensão são adversas para disparo aleatório por tensão, é possível para um semicondutor conduzir incorretamente na direção reversa.

Uma condução reversa de um semicondutor para ângulos de disparo maiores, para os quais nenhuma ação corretiva pode ser feita na prática, pode ser auto-extinta, promovendo uma tensão reversa sobre o referido semicondutor. Sem dúvida, o mesmo semicondutor sofrerá um disparo aleatório por tensão, estando no estado de condução.

### **1.1.4-) Períodos de Condução de Semicondutores para Falhas de Comutação**

Nesta modalidade de falha [2], os sinais de controle para disparar os semicondutores de potência, podem enviar ou não os respectivos sinais e, no caso de envio, o ângulo de fase está adiantado ou atrasado de um valor pré-definido( no caso  $60^\circ$ ).

#### **1.1.4.1-) Avanço do Ângulo de Fase**

Observando a figura 3, os SCRs 3 e 6 são disparados a  $0^\circ$  e, os demais são disparados a  $30^\circ$ . Com isso, conclui-se que, a tensão de saída da ponte( $v_d\alpha$ ), trabalhando como retificadora, tende a aumentar, assim como a corrente  $I_d$ .

#### **1.1.4.2-) Atraso do Ângulo de Fase**

Neste tipo de falha, o ângulo de disparo tende a se afastar de  $\alpha = 0^\circ$ , com um conseqüente distanciamento maior que  $60^\circ$  entre o pulso imediatamente anterior.

Na operação de retificação, a tensão  $v_{d\alpha}$  tende a diminuir, tal como  $I_d$ .

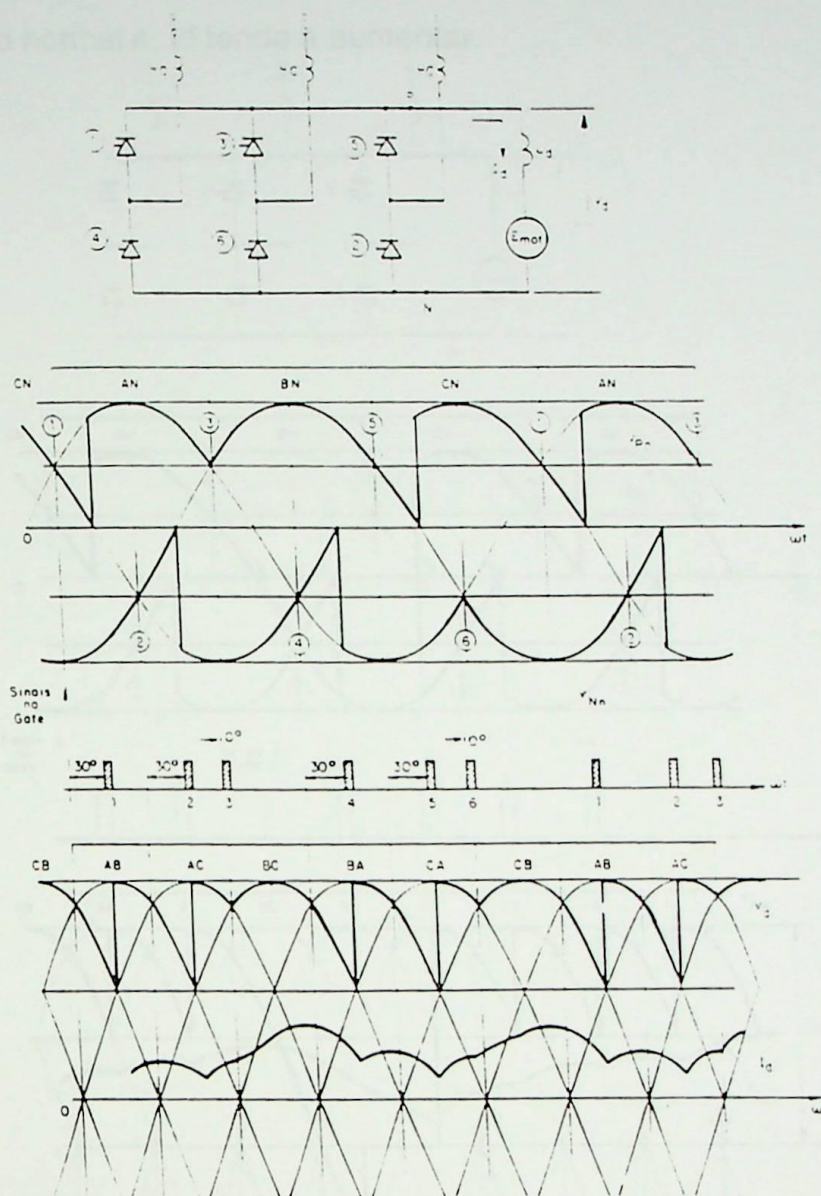


Figura 3 - Avanço de  $\alpha$

### 1.1.4.3-) Ausência de Sinal no Gate

Neste tipo de falha, a tendência de  $v_{d\alpha}$  é diminuir.

A figura 4 mostra as principais formas de onda, quando o pulso de disparo do SCR 3 é inibido. No instante  $t_0$ , ocorre inibição do pulso de gatilho. Conseqüentemente, o SCR 1 continua em condução; já no instante  $t_1$ , o SCR 4 é gatilhado, bloqueando o SCR 2.

Portanto, os SCRs 1 e 4 estão conduzindo, atuando como diodos de livre circulação (free - wheeling). No intervalo  $t_1 - t_2$ , a tensão  $v_{d\alpha}$  será nula ( $I_d$  tende a

cair). Já no instante  $t_2$ , SCR 5 é disparado, causando bloqueio do SCR 1. Logo  $v_d\alpha$  volta ao normal e,  $I_d$  tende a aumentar.

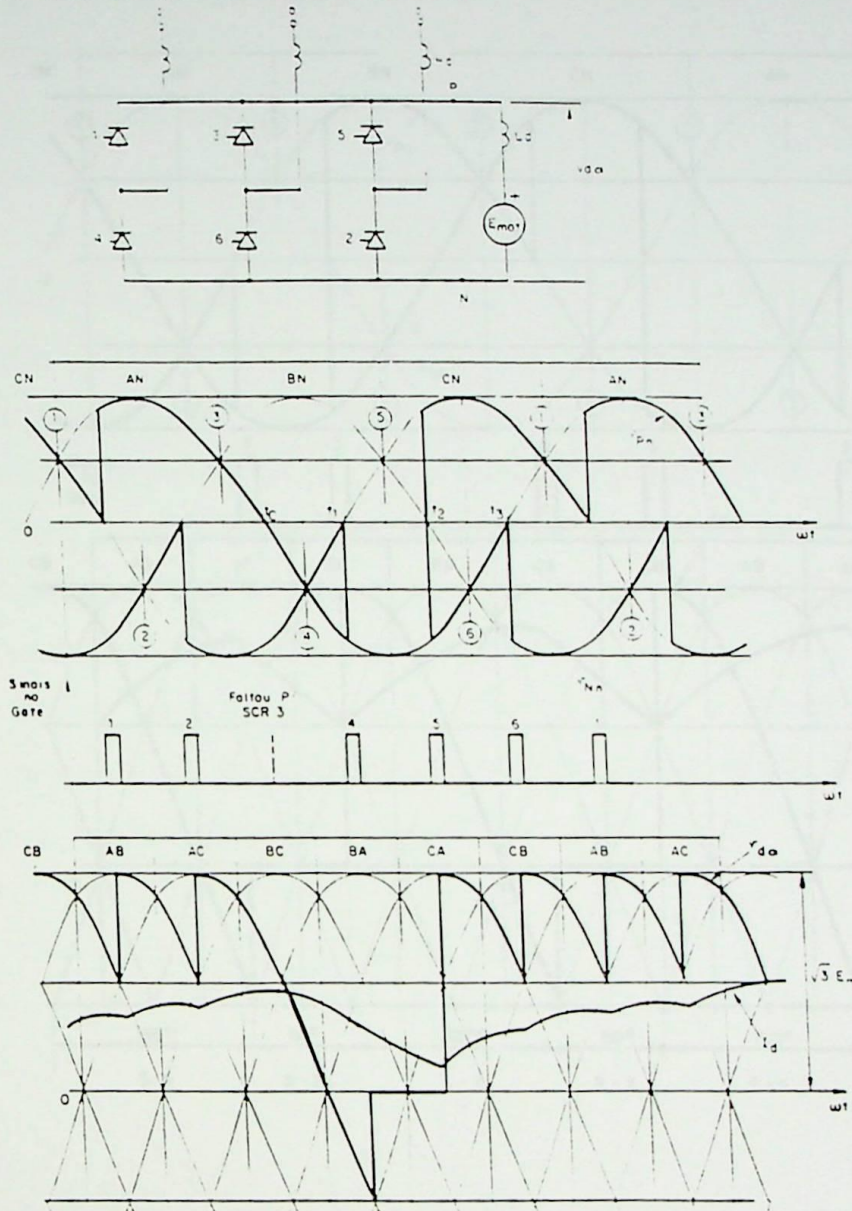


Figura 4 - Ausência de Sinal de Gate ( $\alpha < 90^\circ$ )

### 1.1.4.4-) Ausência de Fase

A figura 5 ilustra as formas de ondas de corrente e tensões na ponte tiristorizada, quando a fase "a" for suprimida.

A ponte se comporta como um circuito bifásico. Também existirão intervalos de tempo, nos quais existem a ação tipo Livre Circulação ("free-wheeling") dos SCR's 2 e 5 e 3 e 6, respectivamente.

Também neste tipo de falha, tanto  $v_d\alpha$  como  $I_d$  terão suas amplitudes diminuídas.

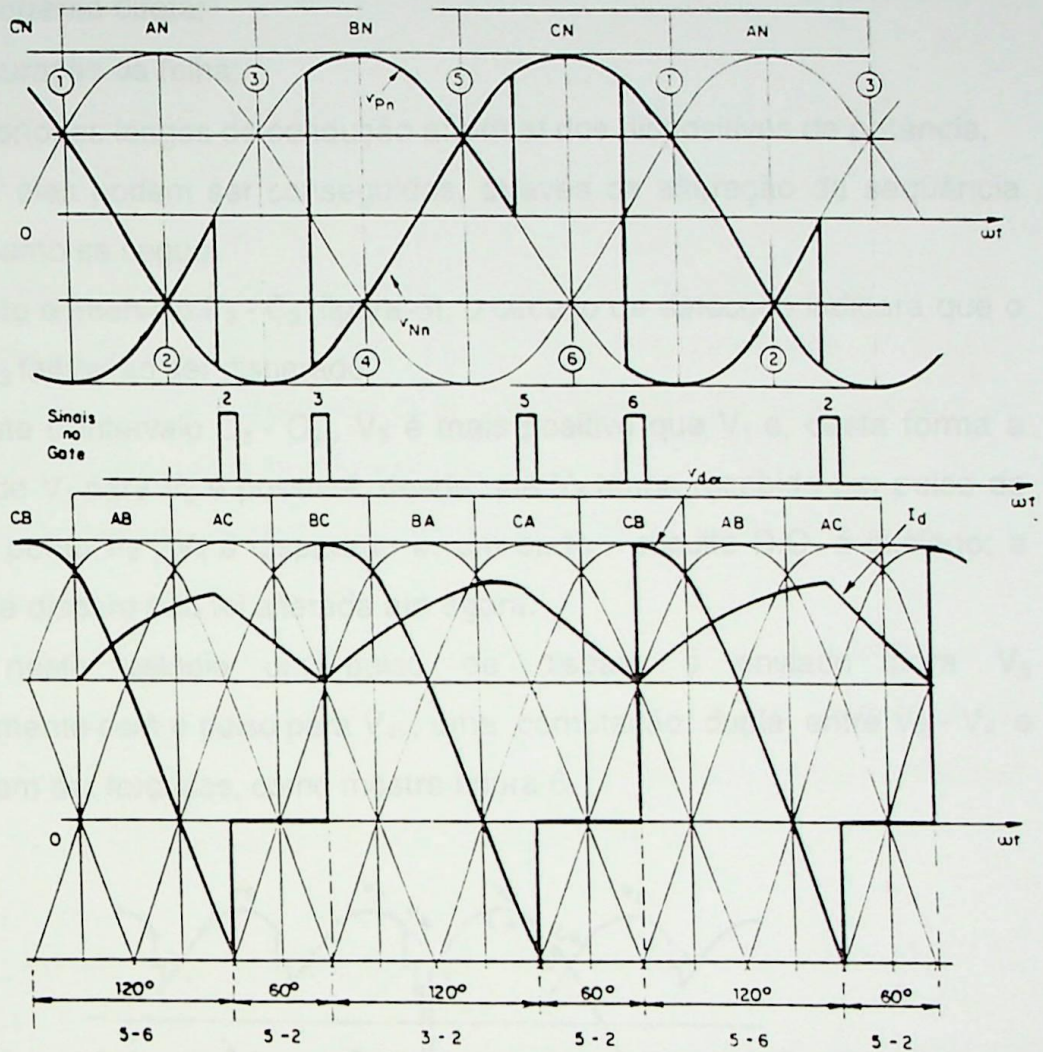


Figura 5 - Ausência de Fases

## 1.2-) Controle da Evolução de Falha em Conversores

O presente tópico mostra que, após a detecção das condições de falha num conversor, pode-se direcionar a seqüência de disparo/bloqueio mais adequada para o desenvolvimento de um tipo de falha .

Para exemplificar esta filosofia, exposta em [3], baseado no grau de severidade de falha sobre os circuitos de potência, escolheu - se para o controle sobre a falha de comutação("misfire"), tanto no inversor como para o retificador.

### 1.2.1-) Controle da Evolução de Falha de Comutação no Inversor

Os objetivos deste controle são:

- reduzir a corrente direta;
- reduzir a duração da falha;
- prevenir períodos longos de condução anormal dos dispositivos de potência.

Todos eles podem ser conseguidos, através da alteração da seqüência de disparo, como se segue.

Durante o intervalo  $P_3 - C_3$  (figura 6), o circuito de detecção indicará que o dispositivo  $V_3$  falhou ao ser disparado.

Durante o intervalo  $C_3 - C_4$ ,  $V_5$  é mais positivo que  $V_1$  e, desta forma a comutação de  $V_1$  para  $V_5$  é possível, desde que  $V_5$  tenha recebido um pulso de disparo. No ponto  $P_4$ ,  $V_4$  é disparado e, um curto - circuito D.C. é iniciado; a seqüência de disparo não foi alterada até agora.

Se, neste estágio um pulso de disparo é enviado para  $V_5$  (simultaneamente com o pulso para  $V_4$ ), uma comutação dupla entre  $V_2 - V_4$  e  $V_1 - V_5$  podem ser forçadas, como mostra figura 6.

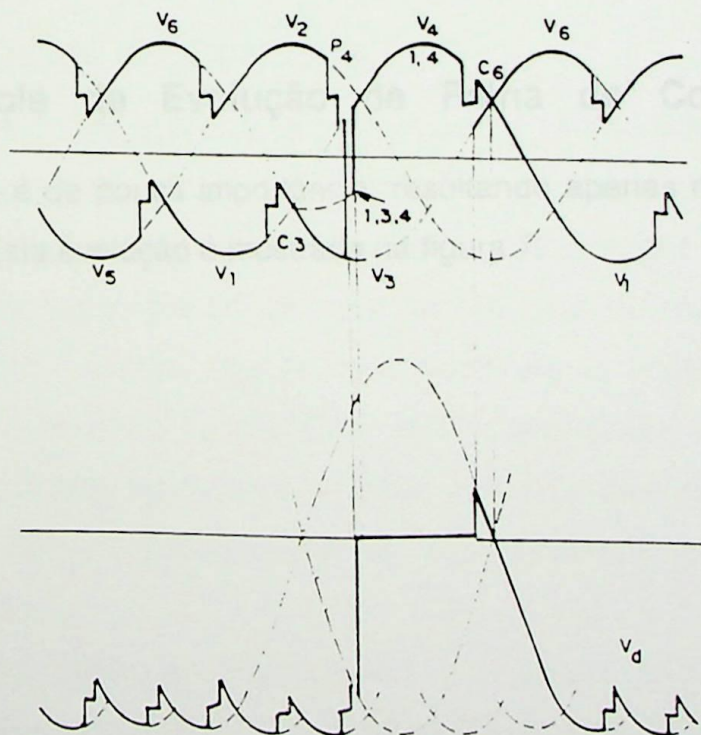


Figura 6 - Evolução Controlada de Falhas de Comutação na Ponte Operando como Inversora

Estes dois pares de dispositivos estão conectados as mesmas fases do transformador e, desta forma as comutações terminarão simultaneamente, levando  $V_4$  e  $V_5$  à condução. Neste ponto, o curto - circuito DC acaba e, a operação normal é finalizada em  $C_5$ .

A duração do curto - circuito DC é agora  $\mu_4$  ao invés de  $(120^\circ + \mu_4)$  com a evolução normal da falha; a corrente direta será então grandemente reduzida.

Após a remoção do curto - circuito DC, a taxa de crescimento de corrente aumenta momentaneamente. No ponto  $P_6$ , a primeira comutação normal ( $V_4$  para  $V_6$ ) ocorre e, este é o ponto no qual a recuperação da falha começa. A duração da falha reduziu de  $300^\circ$  (intervalo  $P_3 - P_6$ ).

$V_1$  e  $V_5$  estão agora conduzindo por  $(180^\circ + \mu)$ , comparado com  $(480^\circ + \mu)$  de condução de  $V_1$  com a evolução normal.

O aumento da corrente direta é muito pequeno com o desenvolvimento controlado da falha; com isso o método pode ser definido como: caso  $V_i$  sofra falha de comutação, deve - se avançar o ângulo  $\beta$  temporariamente e, disparar componente  $V_{i+2}$ , simultaneamente com o componente  $V_{i+1}$ .

### 1.3-) Estratégias de Controle

#### 1.2.2-) Controle da Evolução de Falha de Comutação no Retificador

Esta falha é de pouca importância, resultando apenas na diminuição da corrente direta. Esta evolução é mostrada na figura 7.

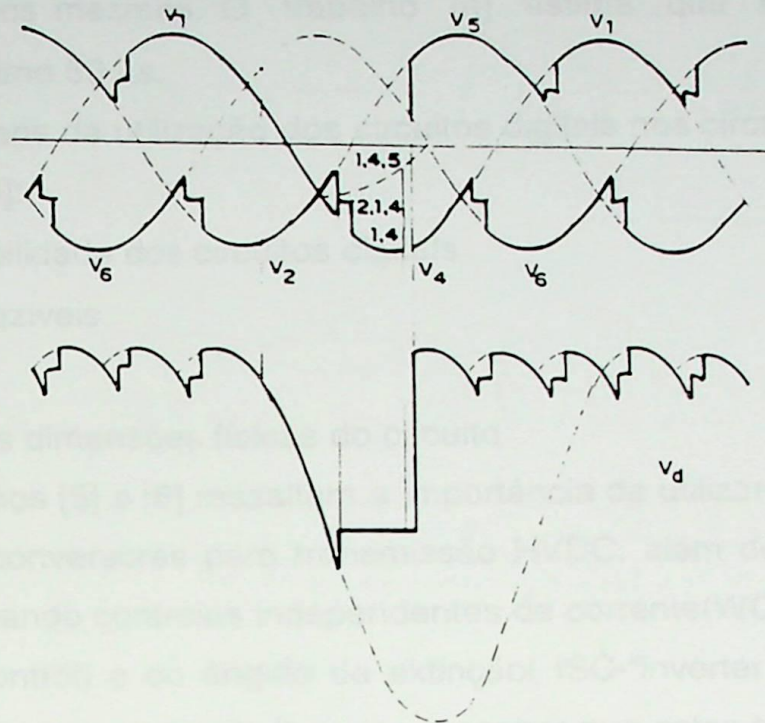


Figura 7 - Evolução Controlada de Falhas de Comutação na Ponte Operando como Retificadora

### 1.3-) Estratégias de Controle

A partir do conhecimento e conseqüente catalogação das falhas em circuitos de potência [1], passou-se agora à fase de elaboração de filosofias do comportamento dos circuitos, assim como formas de proteção dos mesmos.

A primeira estratégia de circuitos voltada para isto é mostrada em [4], onde se utilizam circuitos digitais combinacionais e seqüenciais, contendo somente um tipo de bloco lógico(NOR), tendo como sinais digitais de entrada: estado do componente(conduzindo ou não), pulsos de gatilho dos componentes de potência do conversor e os pontos de "zero crossing"(interseção entre duas tensões de linha).

As saídas deste circuito(contendo 108 blocos NOR), sinalizam uma determinada falha(nível lógico 0), para posterior decisão do sistema, no tocante à indicação e/ou bloqueio do conversor.

Um aspecto que deve ser levado em conta neste tipo de investigação, e o presente trabalho o faz, é o tempo de atraso resultante tanto de características internas de componentes do circuitos, como a filosofia adotada para

implementação dos mesmos. O trabalho [4] estima que haja um atraso global de no máximo 50  $\mu$ s.

As vantagens da utilização dos circuitos digitais nos circuitos de detecção e proteção são [4]:

- grande confiabilidade dos circuitos digitais
- atrasos desprezíveis
- baixo custo
- diminuição das dimensões físicas do circuito

Os trabalhos [5] e [6] ressaltam a importância de utilizar-se blocos digitais no controle de conversores para transmissão HVDC, além de introduzir novos conceitos, mostrando controles independentes de corrente(WOCC - inverter one - way current control) e do ângulo de extinção( ISC-“inverter safety control”) e IOC(“inverter optimum control”). Deve-se ressaltar que estas tecnologias podem ser incorporadas facilmente ao UPS.

No artigo [5] é feito todo embasamento teórico do método, deixando para o outro trabalho todo detalhamento prático[6].

O método proposto é um controle de realimentação negativa, na qual a correção do disparo é proporcional ao desvio dos parâmetros sob controle(corrente, potência, ângulo de extinção) de seu valor nominal.

Na figura 8 é mostrado um diagrama em blocos simplificado da estratégia de controle digital, chamada de DDC( direct digital control)..

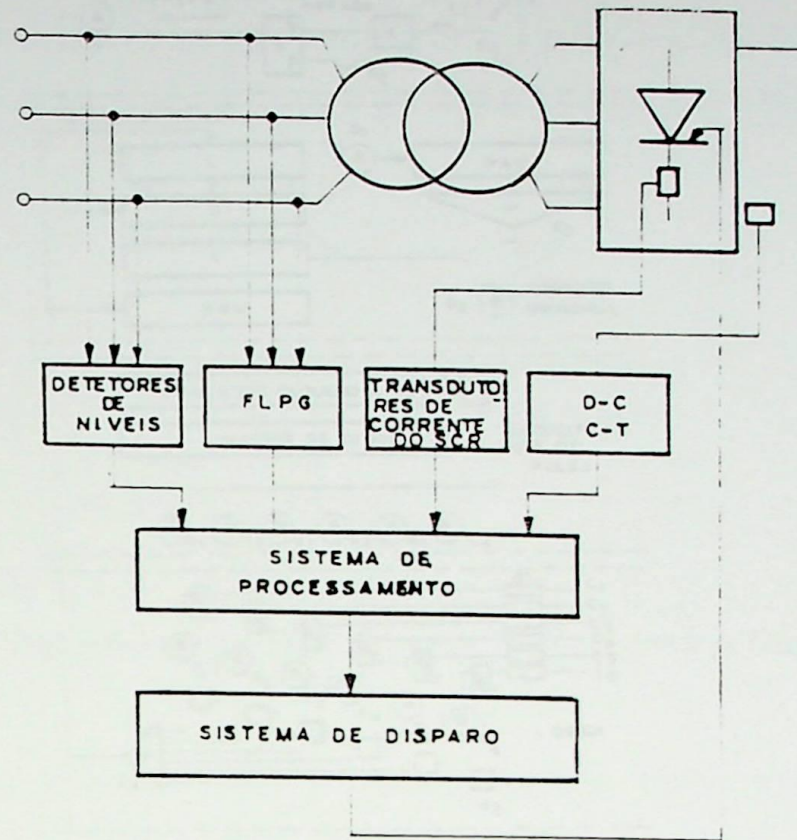


Figura 8 - Diagrama em Blocos do Sistema de Controle

O diagrama em blocos acima inclui: sistema físico (alimentação 3Ø, conversor), sistema de monitoramento e sistema de controle.

O sistema de monitoramento recebe o estado das variáveis de tensão e corrente do sistema físico, enviando-os para o sistema de processamento; este bloco faz toda verificação dos parâmetros e envia ao sistema de disparo dos dispositivos de potência, sinais digitais que irão gerar os sinais de disparo adequados ao funcionamento instantâneo dos dispositivos controlados do sistema de potência.

O sistema de monitoramento é formado por dispositivos discretos, a saber: detetores de "zero crossing" e condução ou não dos dispositivos de potência, transformador de corrente DC e gerador de pulsos.

Já o sistema de processamento é constituído essencialmente por dispositivos digitais (blocos A/D, registradores, comparadores, blocos lógicos básicos, decodificadores)

O circuito simplificado do controle DDC é mostrado na figura 9.

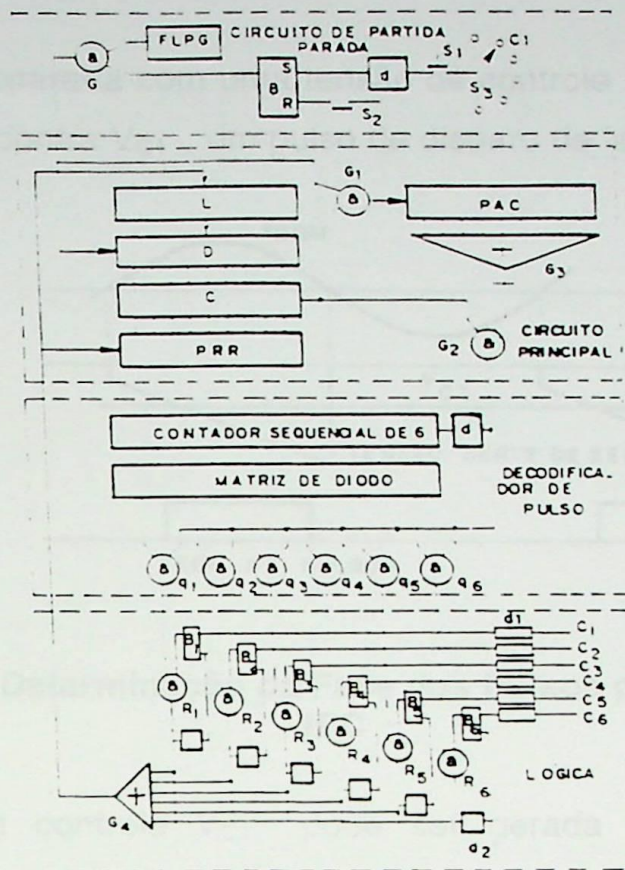


Figura 9 - Gerador de Pulsos de Disparo

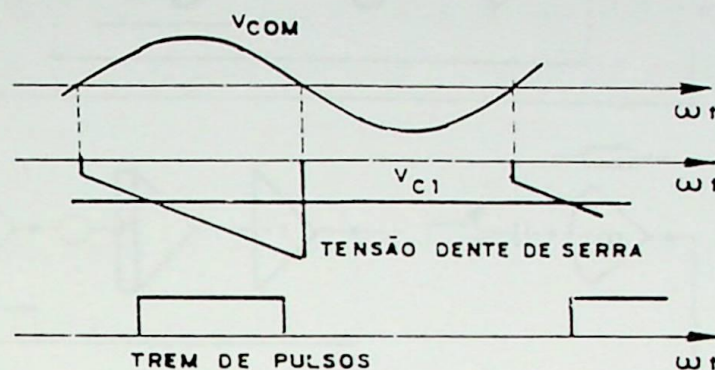
Um outro método de controle dos pulsos, que irá comandar, de forma mais estável e segura, os circuitos de potência é comentado a seguir a partir do trabalho, que é dividido em duas partes. A primeira parte apresenta teoria e circuitos básicos de dois métodos de controle em sistemas HVDC com problemas[7] e, a segunda parte[8] apresenta e comenta os resultados obtidos na prática, em sistemas desbalanceados.

São analisados dois tipos de controle usando trem de pulsos(“grid pulse”): controle de fase individual(IPC- “individual phase control” ) e também o controle utilizando pulsos eqüidistantes(PPC - “equidistant pulses control”).

No controle IPC, o início e final do trem de pulsos é determinado pela interseções entre uma tensão de controle e um sinal dente-de-serra, que possui a mesma largura de sinal do semi - ciclo positivo da tensão de linha. Ainda neste controle, é usado um filtro ativo, com intuito de diminuir oscilações causadas por harmônicas.

A figura 10 mostra a determinação da fase para disparo dos pulsos no método IPC. Uma tensão dente-de-serra é gatilhada na tensão de 0 V da tensão

de comutação e comparada com uma tensão de controle  $V_{C1}$ . Quando a tensão dente - de - serra encontra  $V_{C1}$ , um pulso de disparo de largura  $120^\circ$  é gerado.



**Figura 10 - Determinação da Fase dos Pulsos para o Controle Tipo IPC**

A tensão de controle  $V_{C1}$  pode ser gerada tanto de um circuito controlador de corrente, como de um circuito de extinção

O outro tipo de controle, que é uma variação dos métodos já apresentados anteriormente, permite controlar o conversor não só através de malha fechada, como por um determinado ângulo de controle, o qual é interessante no caso de chaveamento de uma ponte.

Pode-se mostrar a eficácia de ambas as filosofias [8], quando o sistema HVDC está debilitado. O controle IPC e com o controle de extinção são mais adequados para aumentar a estabilidade de sistemas AC debilitados.

Caso ocorram variações de frequência no sistema AC ou quando a estabilidade não for crítica, o método de pulsos equidistantes deve ser usado.

No controle IPC é utilizado filtro ativo, conforme mostra figura 11, com intuito de diminuir oscilações causadas por harmônicas, possuindo as seguintes características:

- máxima atenuação possível de frequências acima daquela utilizada pelo sistema AC (no caso real  $f_R = 50$  Hz);
- frequência fundamental do sistema não é deslocada;
- tempo de "settling" mínimo possível;
- operação independente da amplitude do sinal AC;
- estabilidade por "longtime".



O ângulo  $\alpha$  é medido, via amplificador somador mostrado na figura 13, e enviado a um amplificador de controle. A tensão sobre C serve como valor de referência de  $\alpha$ . A tensão de saída  $V_{c22}$  representa um dos dois componentes que formam a tensão  $V_{c2}$ , a qual controla a fonte de corrente, conforme mostrado abaixo. a outra parcela  $V_{c21}$  é derivada do sistema AC.

Com este sistema de grade de controle é fácil impor limites para o ângulo  $\alpha$ , somente observando a faixa de variação da tensão de controle  $V_{c1}$ .

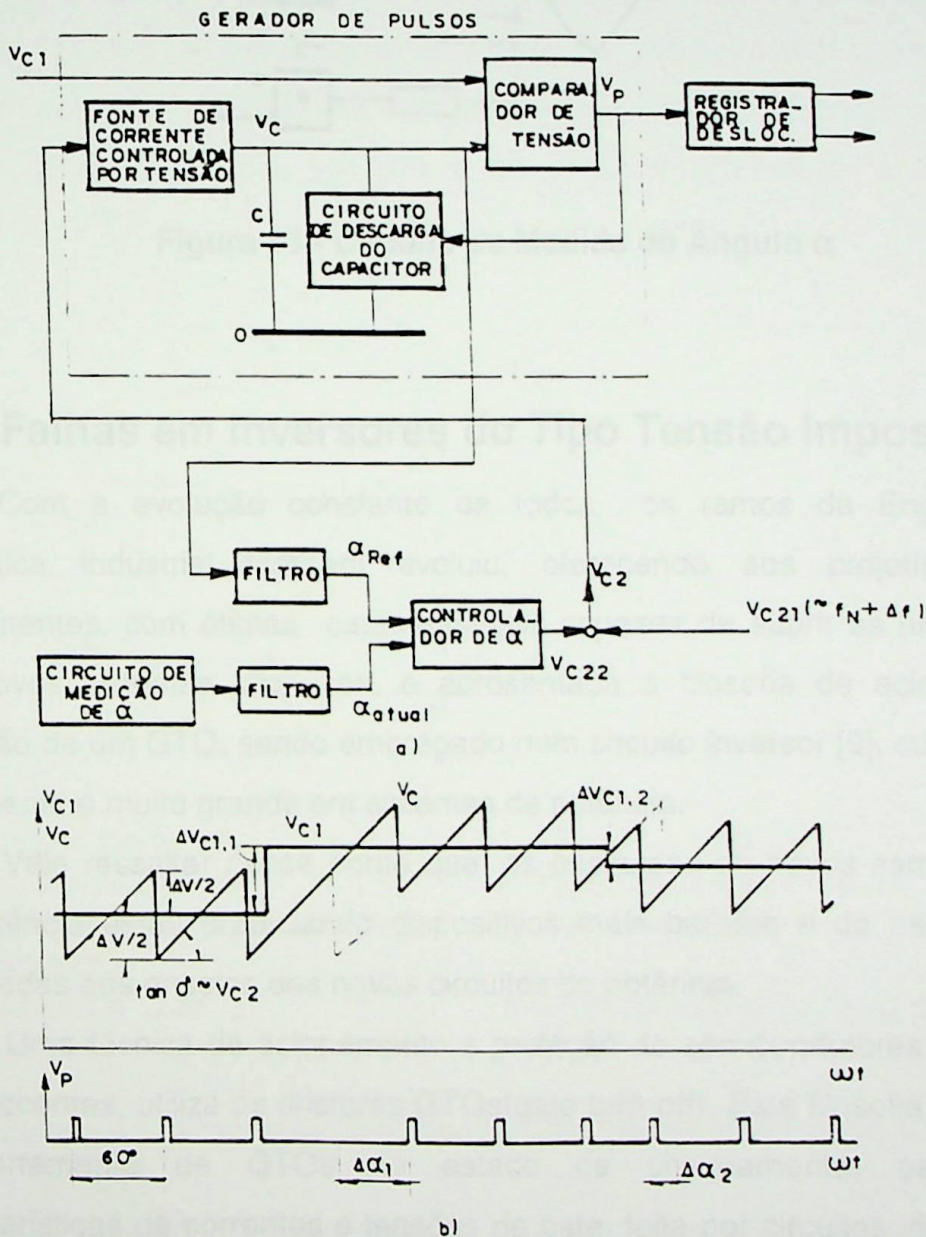


Figura 12 - Sistema de controle de Grade de Pulsos(PPC)  
 a) Diagrama em Blocos  
 b) Determinação da Fase dos Pulsos de Disparo

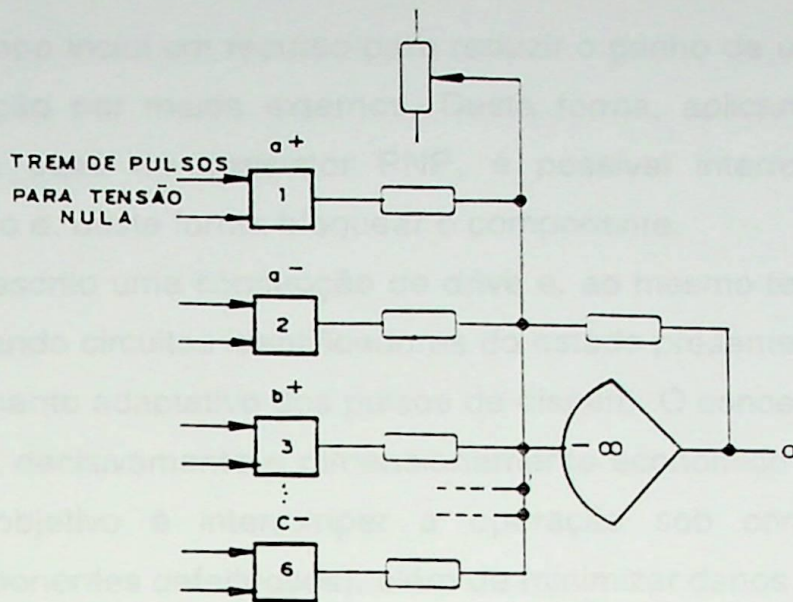


Figura 13 - Circuito de Medida do Ângulo  $\alpha$

### 1.4-) Falhas em Inversores do Tipo Tensão Imposta

Com a evolução constante de todos os ramos da Engenharia, a Eletrônica Industrial também evoluiu, oferecendo aos projetistas novos componentes, com ótimas características, capazes de suprir as necessidades dos novos sistemas. Por isso, é apresentada a filosofia de acionamento e proteção de um GTO, sendo empregado num circuito inversor [9], cuja utilização atualmente é muito grande em sistemas de potência.

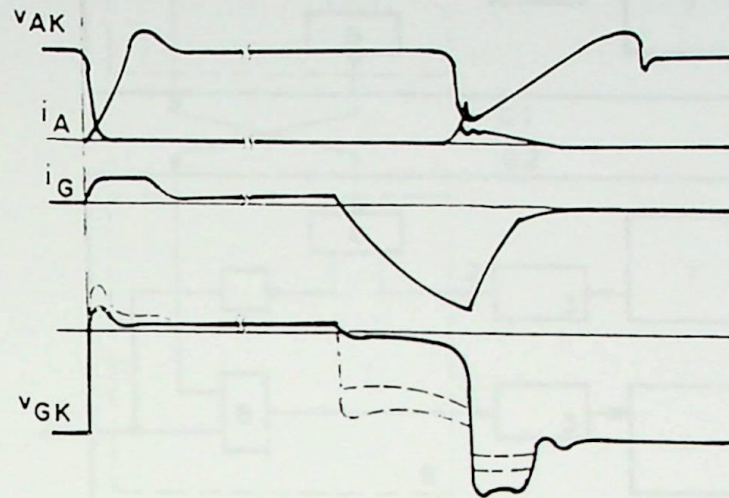
Vale ressaltar nesse ponto que, as pesquisas de novos semicondutores de potência estão propiciando dispositivos mais baratos e de especificações adequadas aos anseios dos novos circuitos de potência.

Uma técnica de acionamento e proteção de semicondutores de potência mais recentes, utiliza os tiristores GTOs (gate turn off). Esta filosofia apresenta o comportamento de GTOs no estado de chaveamento, baseada nas características de correntes e tensões de gate, feita por circuitos identificadores específicos; este comportamento influi na operação de acionamento do semicondutor.

O GTO (Gate Turn Off Thyristor) é uma variação do SCR que, pode ser desligado por um sinal elétrico. Trata-se de um tiristor cuja estrutura foi dimensionada para fornecer maiores velocidades de comutação e que, ao



As características de tensão e corrente observadas sobre o componente , estão na figura 15, sendo que a linha tracejada é a tensão de saída da unidade de disparo do GTO.

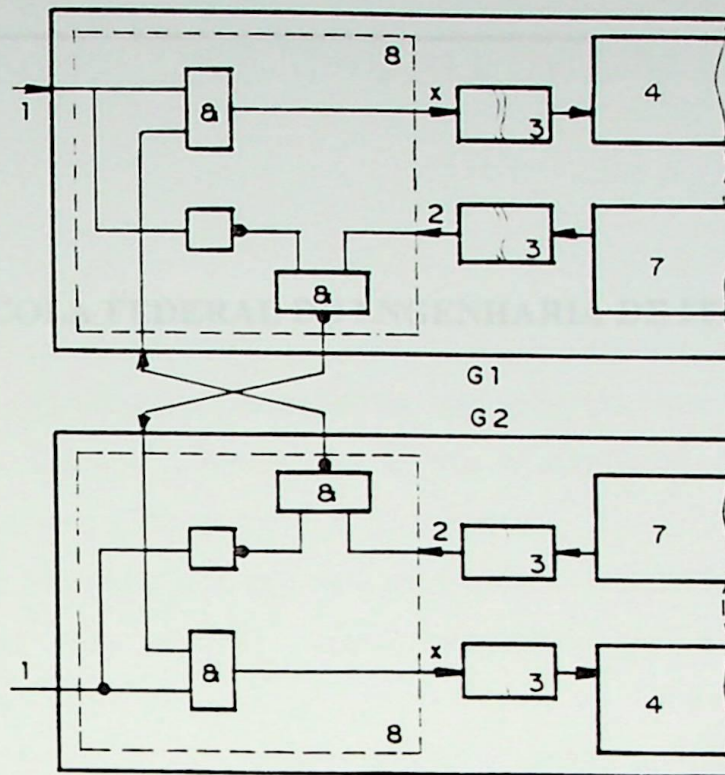


**Figura 15 - Correntes e Tensões do GTO durante Condução e Bloqueio**

O bloco Intertravamento Adaptativo torna possível operar o inversor com mínimo de tempo de intertravamento, além de evitar um curto - circuito na conexão de tensão DC, no caso de falha tanto da unidade de disparo ou do próprio componente.

O intertravamento se processa pela recepção por parte dos respectivos blocos de cada componente, de sinais de estado outro GTO, isto é, um GTO não conduz enquanto não receber sinal para tal e, o sinal de que o outro GTO recebeu o sinal de bloqueio e permanece nesse estado.

A figura 16, mostra o diagrama em blocos do bloco de intertravamento.



G1 – Unidade de Disparo do Gate  
 1 - Sinal de Comando  
 3 – Isoladores de Sinal  
 7 – Detetor de Estado

G2 – Unidade de Disparo do Gate  
 2 – Sinal de Estado  
 4 – Gerador de Pulsos do Gate  
 8 – Intertravamento Adaptativo

Figura 16 - Diagrama em Blocos do Bloco de Intertravamento Adaptativo

A lógica de disparo de proteção visa proteger os GTOs, quando o tempo de armazenamento (“ storage time”) do componente, intervalo de tempo entre a ocorrência do sinal de bloqueio e o sinal de estado do GTO, exceder a um valor pré - determinado. Esta proteção é particularmente adequada, para GTOs de baixo custo, particularmente na ocorrência de curto - circuito numa ponte, pois com alto  $di/dt$ , o comando de bloqueio leve um certo tempo para atuar.

Outro semiconductor recente, bastante utilizado nos circuitos de potência é o transistor de potência IGBT(insulated gate bipolar transistor). Existem circuitos relativos a seu disparo e acionamento, sendo um deles comentado no capítulo 3.



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## CAPÍTULO 2

### DESCRIÇÃO DE UM UPS

Figura 17 - Diagrama em Blocos de um UPS Típico

Deste diagrama em blocos, podemos observar que o UPS é composto basicamente, de três elementos: Retificador, Banco de Baterias e Inversor.

O Retificador, converte em tensão contínua a tensão alternada da rede elétrica e desta forma, graças do barramento DClink DCI, pode-se carregar o Banco de Baterias que é o elemento armazenador de energia do UPS. O barramento DC tem duas funções: a primeira é alimentar o Banco de Baterias e

## Capítulo 2 - DESCRIÇÃO DE UM UPS

### 2.1-) Definição

O UPS(Uninterruptible Power Supply) é um equipamento que tem como função principal o suprimento permanente de energia elétrica a sistemas sensíveis a falta de energia elétrica, de forma automática, de forma que o condicionamento da energia fique livre de harmônicos.

Muito utilizado em instalações onde não possa ocorrer falta ou oscilação de energia elétrica, tais como : processamento de dados, hospitais, comunicação, aviação, fábricas , etc.

O UPS, além de evitar que sejam perdidos dados no caso de uma falha de energia elétrica, também protege o equipamento contra descargas estáticas e variações da rede elétrica, prolongando a vida útil do equipamento nele ligado.

A figura 17 a seguir, ilustra o diagrama em blocos deste equipamento. Mais adiante, será ilustrado os modos de operação.

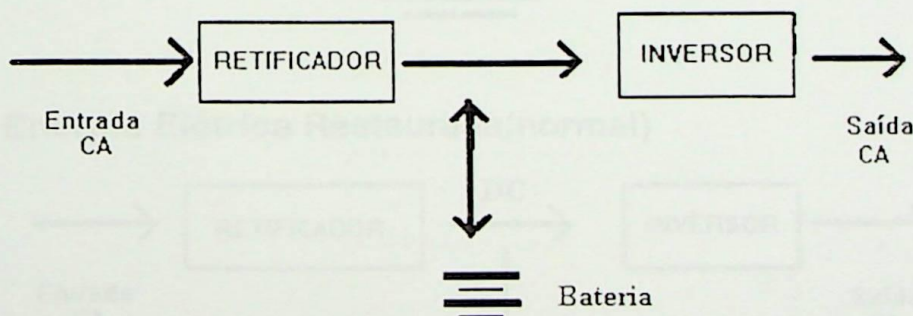


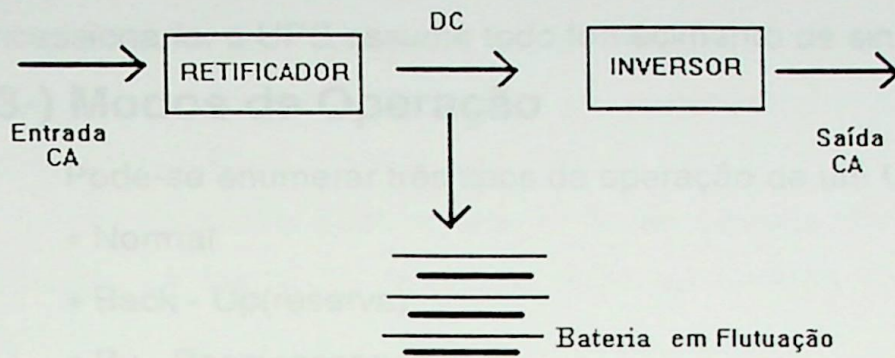
Figura 17 - Diagrama em Blocos de um UPS Típico

Deste diagrama em blocos, podemos observar que o UPS é composto, basicamente, de três elementos: Retificador, Banco de Baterias e Inversor.

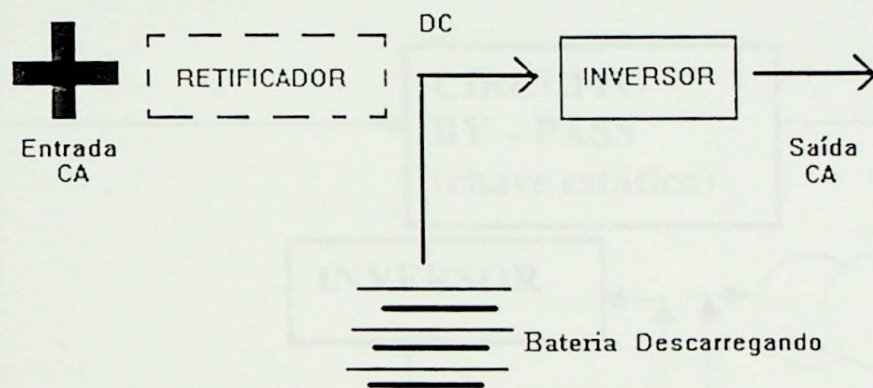
O Retificador, converte em tensão contínua a tensão alternada da rede elétrica e desta forma, através do barramento DC(link DC), pode-se carregar o Banco de Baterias que é o elemento armazenador de energia do UPS. O barramento DC tem dois objetivos: o primeiro é alimentar o Banco de Baterias e,

o segundo é alimentar o Inversor que reconverte em tensão alternada a tensão contínua do barramento DC. A figura 18 apresenta esses modos de operação.

**Operação Normal**



**Falha na Rede Elétrica**



**Energia Elétrica Restaurada(normal)**

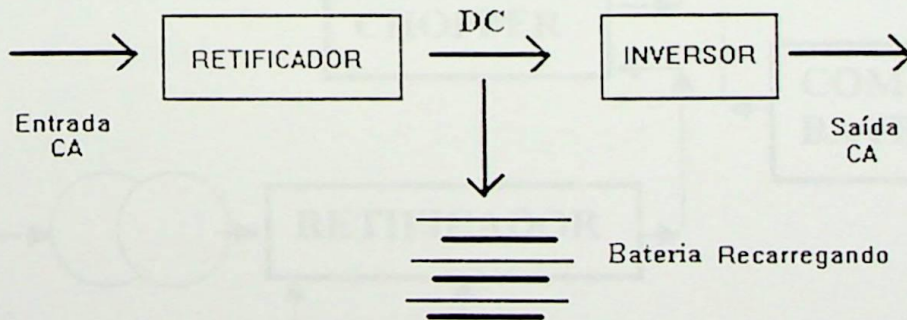


Figura 18 - Modos de Operação do UPS

**2.2-) Classificação**

UPS está dividido em três categorias: Linha Direta("On-Line"), Espera("Off-Line ou Stand-By") ou Linha Interativa("Interactive Line").

No tipo Linha Direta, o equipamento só é alimentado pela bateria.

Já no tipo Stand-by, alimentação do equipamento é feita pela rede elétrica, passando pela bateria, em caso de falha elétrica da rede.

A terceira categoria, é um meio-termo entre as anteriores. Neste caso, o Inversor trabalha em paralelo com a rede e, ocorrendo falha elétrica da concessionária, o UPS assume todo fornecimento de sinal AC.

## 2.3-) Modos de Operação

Pode-se enumerar três tipos de operação de um UPS:

- Normal
- Back - Up(reserva)
- By - Pass(passagem)

Na figura 19, é mostrado um circuito simplificado de um UPS em que serão baseadas as explicações de cada modo de operação.

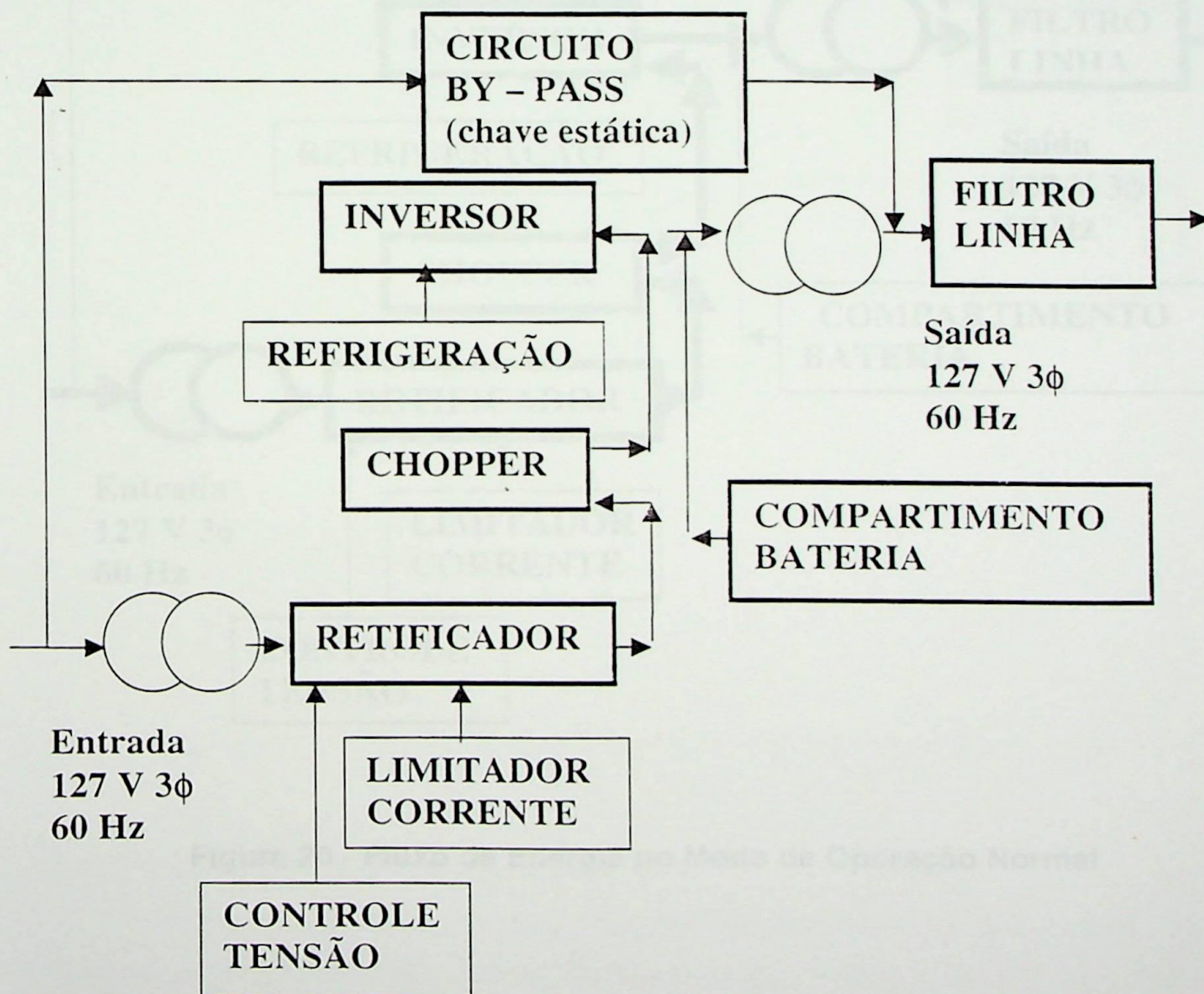


Figura 19 - Circuito Simplificado de um UPS

Figura 19 - Circuito Simplificado de um UPS

### 2.3.1-) Modo de Operação Normal

Neste modo de operação, o retificador RF converte sinal AC vindo da concessionária em sinal DC, para ser aplicada no Inversor INV, através de um chopper(omitido do circuito).

As baterias, são constantemente carregadas durante este modo de operação. A figura 20 mostra o fluxo de energia deste modo de operação, através da seta em negrito.

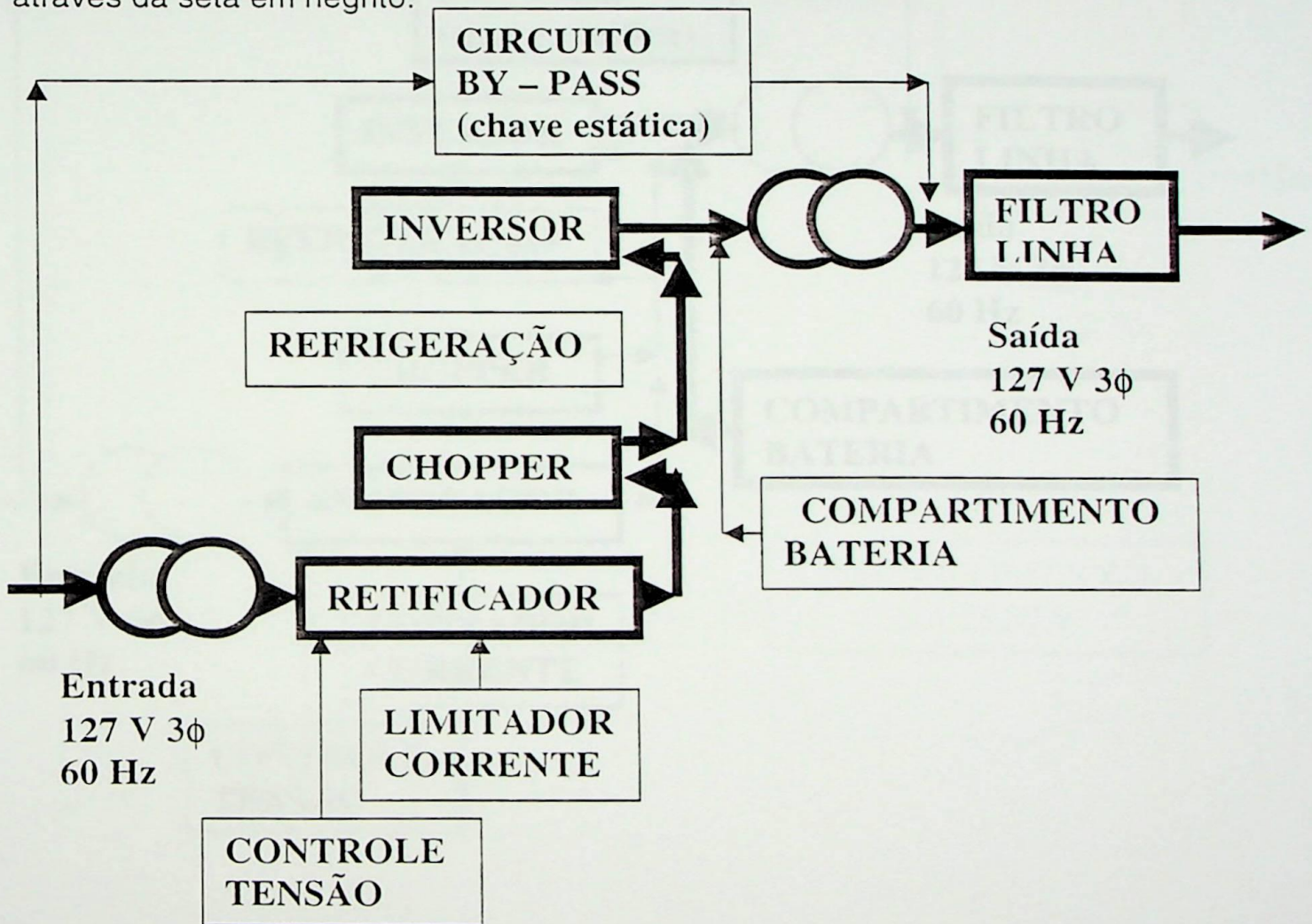


Figura 20 - Fluxo de Energia no Modo de Operação Normal

### 2.3.2-) Modo de Operação Back - Up

Quando a rede AC, por quaisquer motivos, for interrompida, é o grupo de baterias que deverá fornecer potência à carga, conforme indicado pela seta em negrito na figura 21. Este fornecimento deve ser interrompido, assim que a tensão da bateria diminuir a partir de um certo valor de tensão.

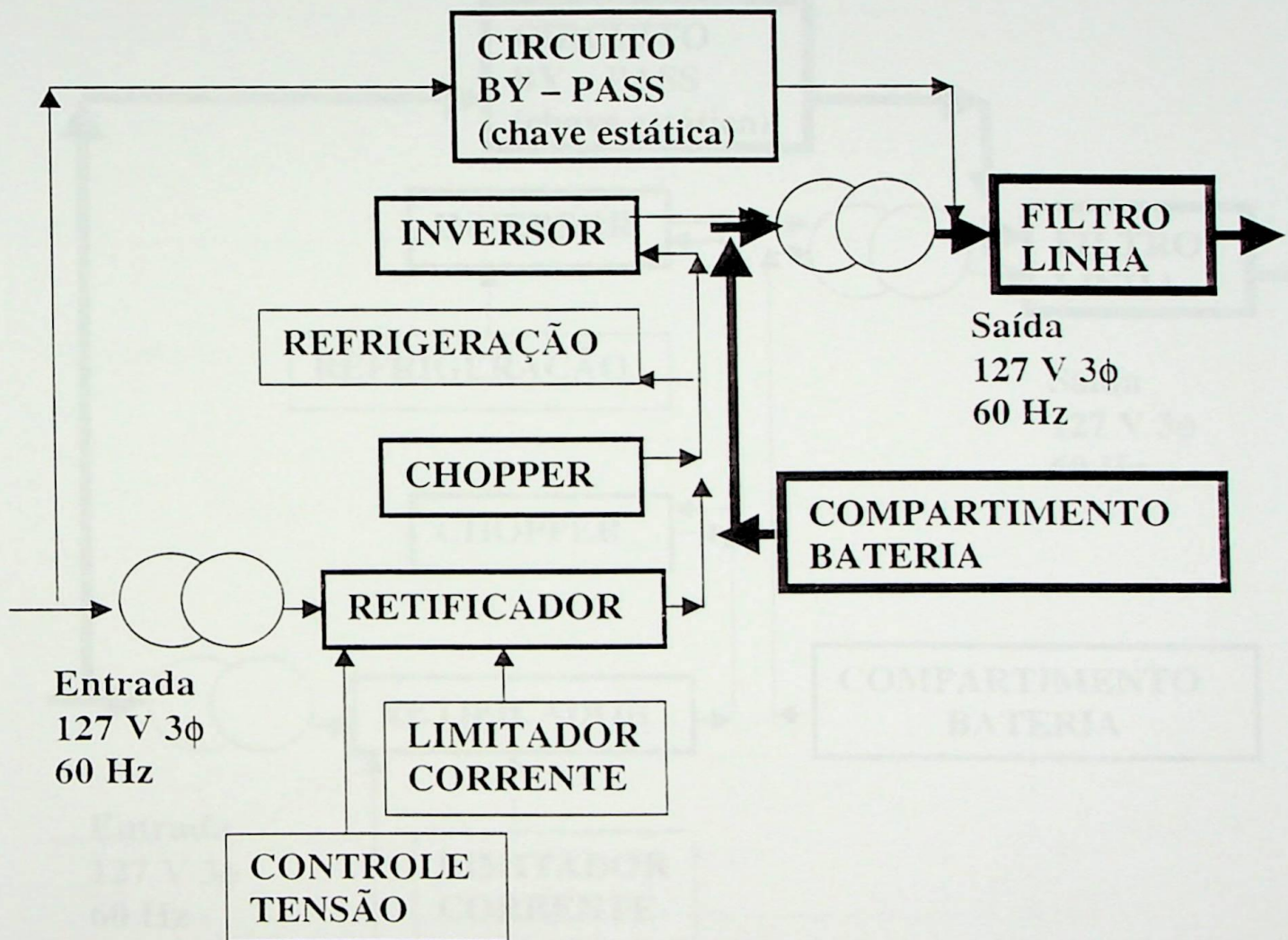


Figura 21 - Fluxo de Energia no Modo de Operação Back - Up

### 2.3.3-) Modo de Operação By - Pass

Quando for necessário uma manutenção em qualquer dos blocos que formam o UPS (Retificador, Banco de Baterias ou Inversor), pode-se ativar o contato  $R_{y2}$ , de modo que o fluxo de potência flua, diretamente, da concessionária à carga, conforme mostra a seta em negrito da figura 22.

Quando for necessário uma manutenção em qualquer dos blocos que formam o UPS (Retificador, Banco de Baterias ou Inversor), pode-se ativar o contato  $R_{y2}$ , de modo que o fluxo de potência flua, diretamente, da concessionária à carga, conforme mostra a seta em negrito da figura 22.

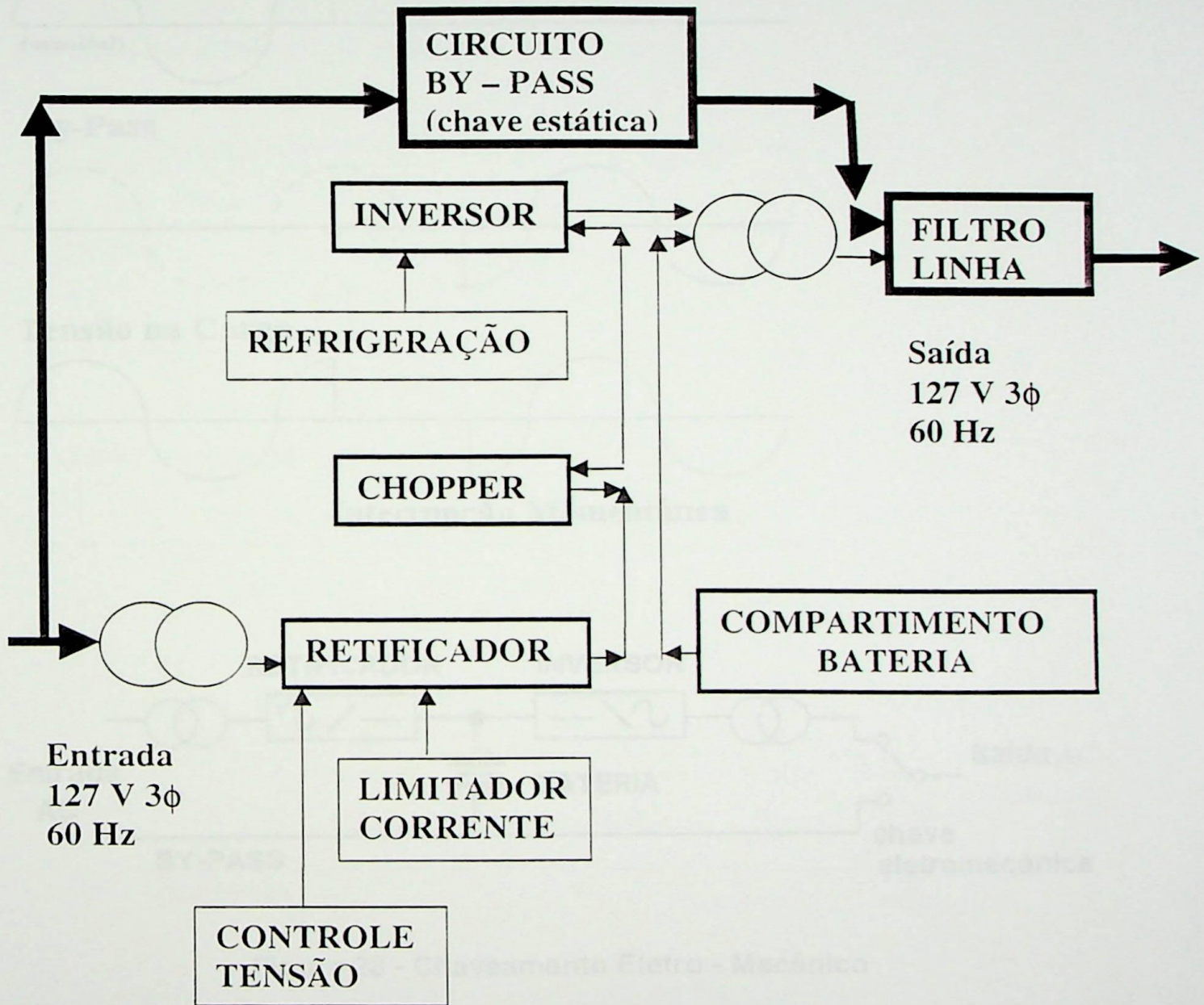


Figura 22 - Fluxo de Energia no Modo de Operação By – Pass

- Transferência Eletro-Mecânica Reversa: sistema barato, interrupção momentânea da tensão na carga durante chaveamento. A figura 23 mostra tanto as formas de onda, como o respectivo diagrama em blocos.

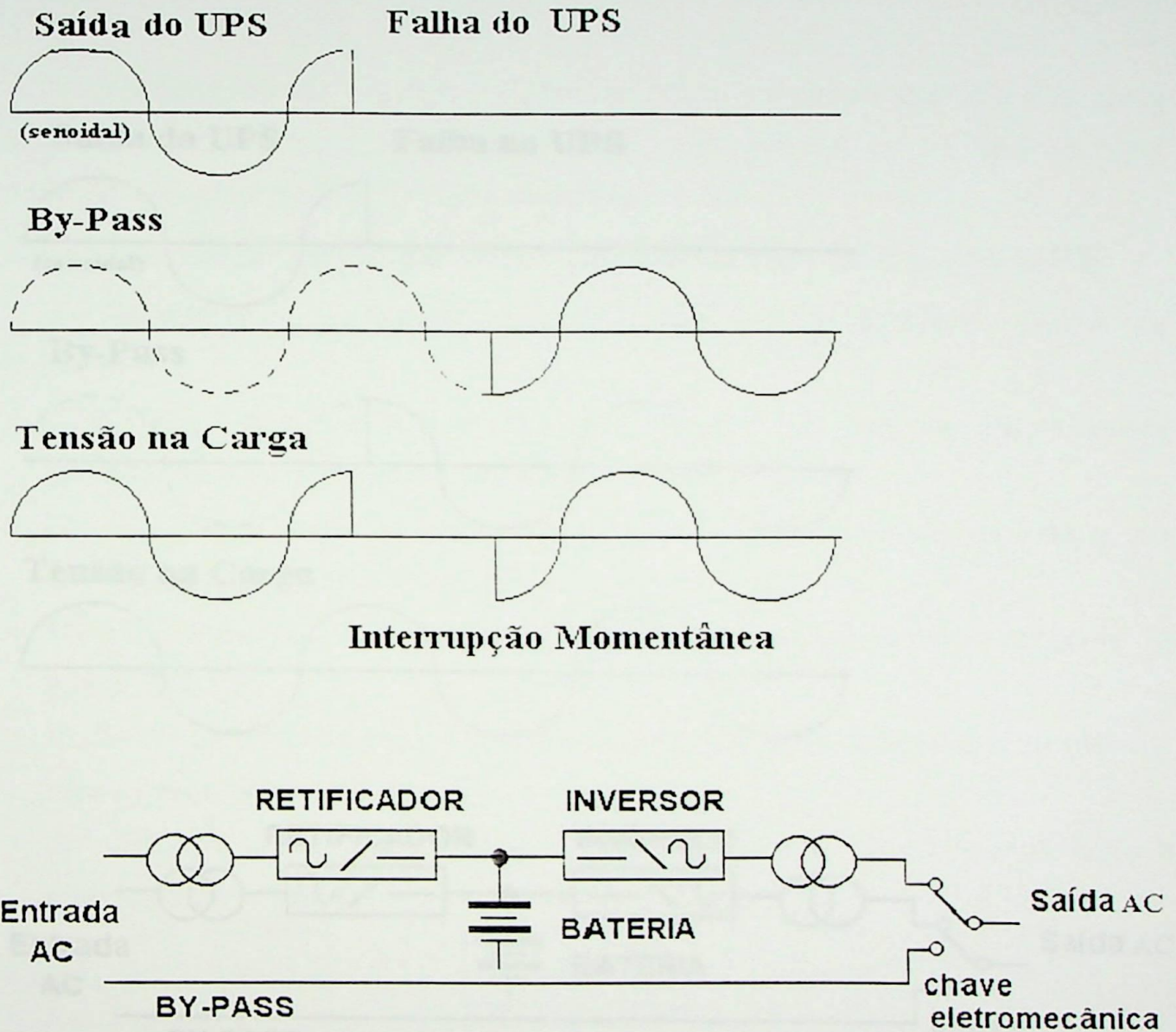


Figura 23 - Chaveamento Eletro - Mecânico

- Transferência Estática Reversa: utiliza como elemento chaveador chaves estáticas(SCRs em anti - paralelo). A figura 24 mostra tanto as formas de onda, como o respectivo diagrama em blocos.

- Transferência Estática Reversa: utiliza como elemento chaveador chaves estáticas(SCRs em anti - paralelo). A figura 24 mostra tanto as formas de onda, como o respectivo diagrama em blocos.

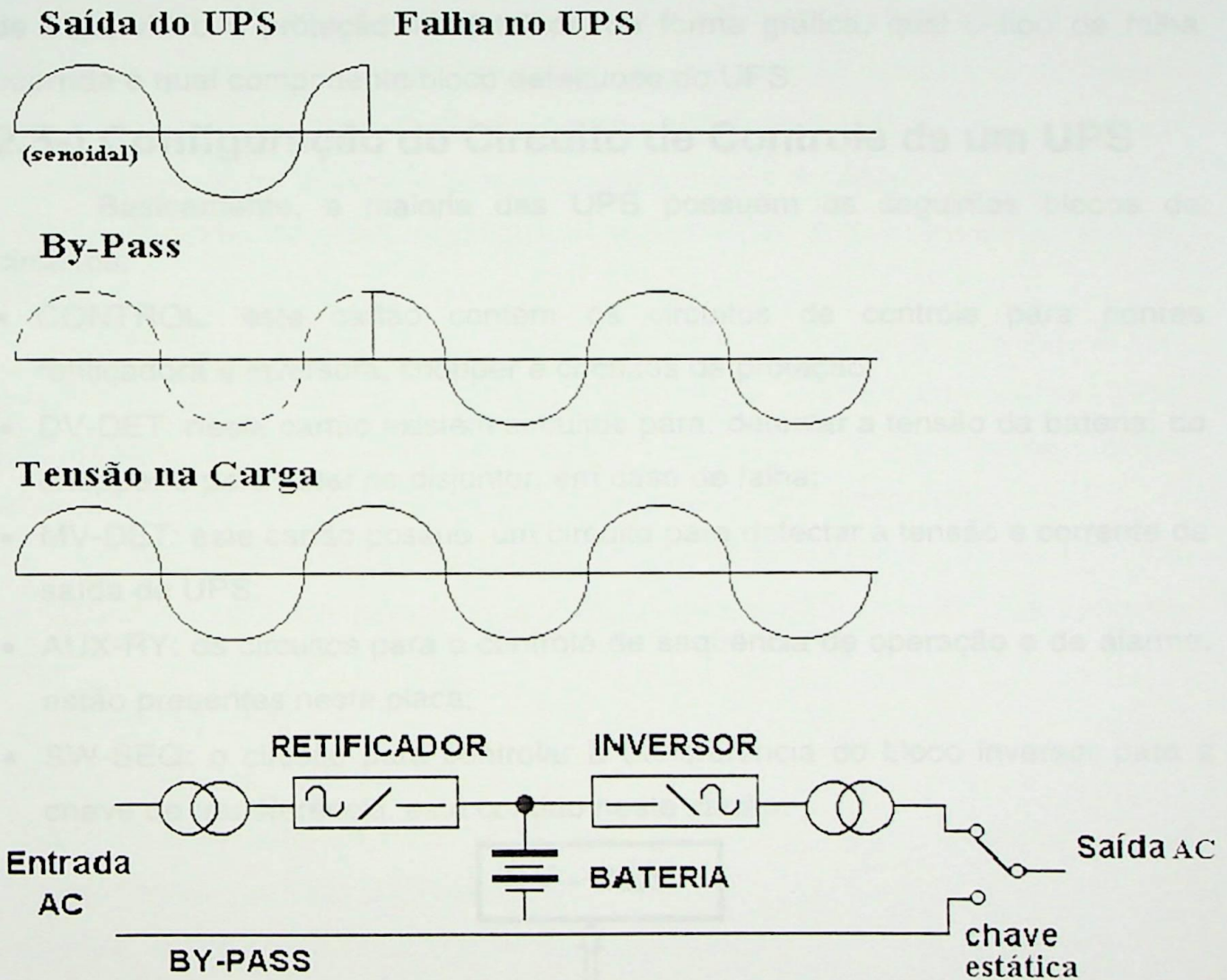


Figura 24 - Chaveamento Estático

## 2.4-) Indicações Visuais de um UPS

De maneira geral, podemos encontrar as seguintes indicações de sinalização no painel frontal de um UPS:

- AC MAIN: indica se existe alimentação elétrica da rede CA.
- OUT PUT: indica existência ou não de tensão sobre a carga;
- INVERTER: indica se o bloco Inversor está funcionando adequadamente;
- FAULT: indica atuação da proteção sobre o circuito do UPS(sobrecorrente, sobrecarga, sobretensão, sobreaquecimento e outros defeitos);

As duas últimas indicações, fazem parte do presente trabalho a estratégia de supervisão e proteção irá detalhar, de forma gráfica, qual o tipo de falha ocorrida e qual componente/bloco defeituoso do UPS.

## 2.5-) Configuração do Circuito de Controle de um UPS

Basicamente, a maioria das UPS possuem os seguintes blocos de circuitos:

- CONTROL: este cartão contém os circuitos de controle para pontes retificadora e inversora, chopper e circuitos de proteção;
- DV-DET: neste cartão existem circuitos para: detectar a tensão da bateria, do chopper e para atuar no disjuntor, em caso de falha;
- MV-DET: este cartão possui um circuito para detectar a tensão e corrente de saída do UPS;
- AUX-RY: os circuitos para o controle de seqüência de operação e de alarme, estão presentes nesta placa;
- SW-SEQ: o circuito para controlar a transferência do bloco inversor para a chave de transferência, está contido neste cartão.

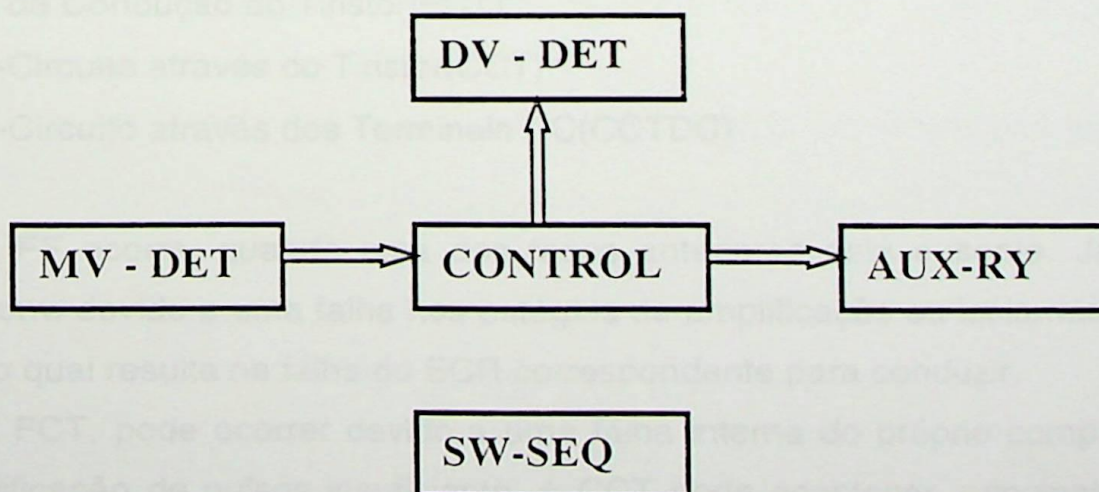


Figura 25 - Configuração do Circuito de Controle

## 2.6-) Falhas Específicas dos Blocos Retificador e Inversor de uma UPS

### 2.6.1-) Falhas Específicas do Bloco Retificador

A ponte trifásica tiristorizada, na prática pode apresentar um funcionamento diferente do esperado pela teoria. Uma das causas, pode ser o circuito de controle do ângulo de disparo, que não libera os pulsos corretamente defasados de  $60^\circ$ . Um ou mais pulsos podem estar defasados entre si de um ângulo menor que  $60^\circ$  (avanço de gate) ou, maior que  $60^\circ$  (atraso de gate), provocando um aumento ou diminuição da tensão de saída da ponte.

Uma outra falha devido ao circuito de controle do ângulo de disparo, é a ausência de sinal de gate, significando que o SCR que o receberia não irá conduzir.

Também o próprio SCR pode ficar em curto-circuito, devido a um aquecimento excessivo, por exemplo. Neste caso, existirá uma corrente de curto-circuito, entre a fase que alimenta o SCR e as outras fases, correspondentes aos SCR's próximos, que deveriam ser disparados.

Do que foi exposto, pode-se agrupar [10], genericamente, as falhas sobre um circuito de potência funcionando como retificador ( $0^\circ < \alpha < 90^\circ$ ), da seguinte forma:

- Falta de Fase (FF)
- Falha de Pulsos no Gate (FPG)
- Falha de Condução do Tiristor (FCT)
- Curto-Circuito através do Tiristor (CCT)
- Curto-Circuito através dos Terminais DC (CCTDC)

A FF ocorre, quando uma das fases anteriores está ausente. Já FPG pode ocorrer devido a uma falha nos estágios de amplificação ou isolamento dos pulsos, o qual resulta na falha do SCR correspondente para conduzir.

A FCT, pode ocorrer devido a uma falha interna do próprio componente ou amplificação de pulsos insuficiente. A CCT pode acontecer, principalmente, devido a uma falha do dispositivo ou, um curto-circuito acidental entre ânodo e cátodo.

Já CCTDC, pode ocorrer em duas situações:

- CCTDC Antes do Indutor de Alisamento: devido a esta falha, a corrente cresce quase instantaneamente e, eventualmente, pode abrir o fusível, colocado em série com os dispositivos em condução. Caso os pulsos de gatilhamento dos SCRs anteriores estejam desabilitados, fusíveis desses também abrirão. Notar que, para conversores de grande potência, os fusíveis dos semicondutores, os quais estão conectados em série com os tiristores são bastante caros( em torno de 10% do preço do semicondutor de potência).
- CCTDC Após Indutor de Alisamento: um valor alto de indutância de filtro não permite que a corrente cresça abruptamente, mesmo numa situação de curto-circuito. Se o conversor é operado com um laço de corrente, um controlador projetado adequadamente, pode atrasar o ângulo  $\alpha$  para um valor seguro, protegendo desta forma o conversor.

Todas as falhas citadas acima, causam uma redução na tensão e potência de saída e, também provocam um desbalanceamento da carga, ocasionando um aumento nos componentes harmônicos da corrente de entrada.

### 2.6.2-) Falhas Específicas do Bloco Inversor

Já para o circuito Inversor, para facilitar a compreensão e visualização, tabulou-se a seguir, as principais falhas, possíveis causas e soluções em Inversores.

As tabelas a seguir irão mostrar, de forma resumida, as principais falhas de um bloco inversor típico de um UPS.

A tabela 1 procura relacionar as principais falhas em Inversores tipo VSI.

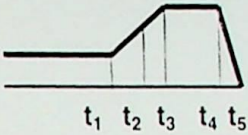
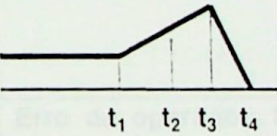
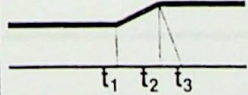
ITEM	CAUSA	EVOLUÇÃO DA SOBRECORRENTE	CARACTERÍSTICAS
Curto-Circuito do lado DC	<p>Operação inadequada do circuito de controle do Inversor tipo ponte e dispositivo defeituoso do circuito</p> <p>Circuito chopper defeituoso, diodo ou circuito de alisamento defeituosos.</p>	<p>Corrente normal</p>  <p><math>t_1</math> <math>t_2</math> <math>t_3</math> <math>t_4</math> <math>t_5</math></p> <p><math>t_4</math> : Ponto "cut - off" do transistor</p> <p><math>t_1</math> : Ponto de falta</p> <p><math>t_2</math> : Nível de ajuste de sobrecorrente</p> <p><math>t_3</math> : Ponto no Qual a corrente de curto - circuito é controlada por <math>h_{fe}</math></p>	<p>Se a tensão DC for <math>E_s</math> e a indutância da linha de curto - circuito for <math>L</math>, a elevação da corrente é <math>di/dt = E_s/L</math>, que é várias centenas de amperes em vários <math>\mu s</math>.</p> <p>A corrente entra na região ativa em <math>t_2</math>, logo <math>t_{stq}</math> no tempo de "cut - off" é quase zero.</p>
Curto-circuito da Carga	<p>Curto da camada ou saturação do transformador de saída.</p>	<p>Corrente normal</p>  <p><math>t_1</math> <math>t_2</math> <math>t_3</math> <math>t_4</math></p> <p><math>t_1, t_2</math> : Idem ao acima</p> <p><math>t_3</math> : Tempo de "cut - off" do transistor</p>	<p>A indutância da linha de curto - circuito é maior do que a acima. Quando o retardo do detetor é pequeno, a corrente não atingirá <math>I_s</math>.</p> <p>A corrente entra na região ativa em <math>t_2</math>, logo <math>t_{stq}</math> no tempo de "cut - off" é quase zero.</p>
Sobrecarga	<p>Operação inadequada</p> <p>Carga defeituosa</p>	<p>Corrente normal</p>  <p><math>t_1</math> <math>t_2</math> <math>t_3</math></p>	

Tabela 1 - Causas de Sobrecorrentes e Características

A tabela 2, mostra as causas de sobrecorrente num Inversor, assim como o caminho dessa corrente indesejável no circuito.

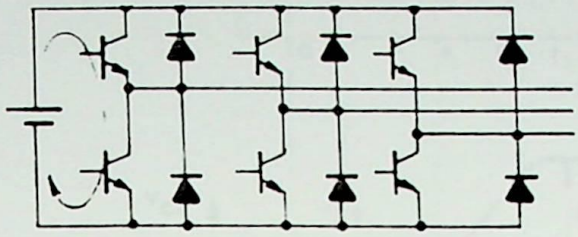
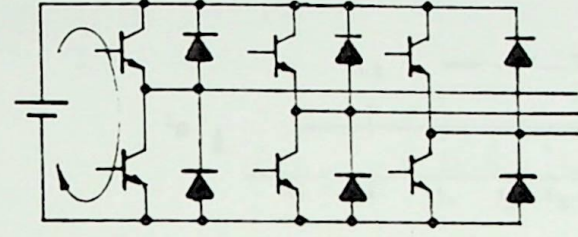
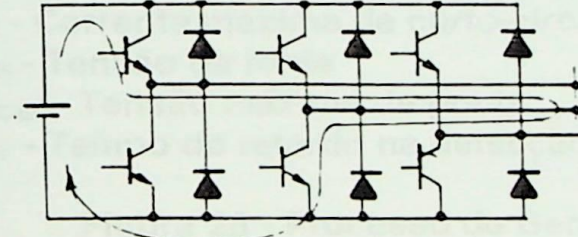
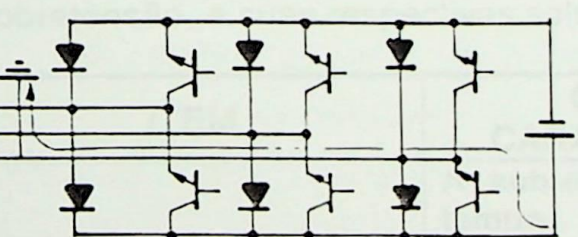
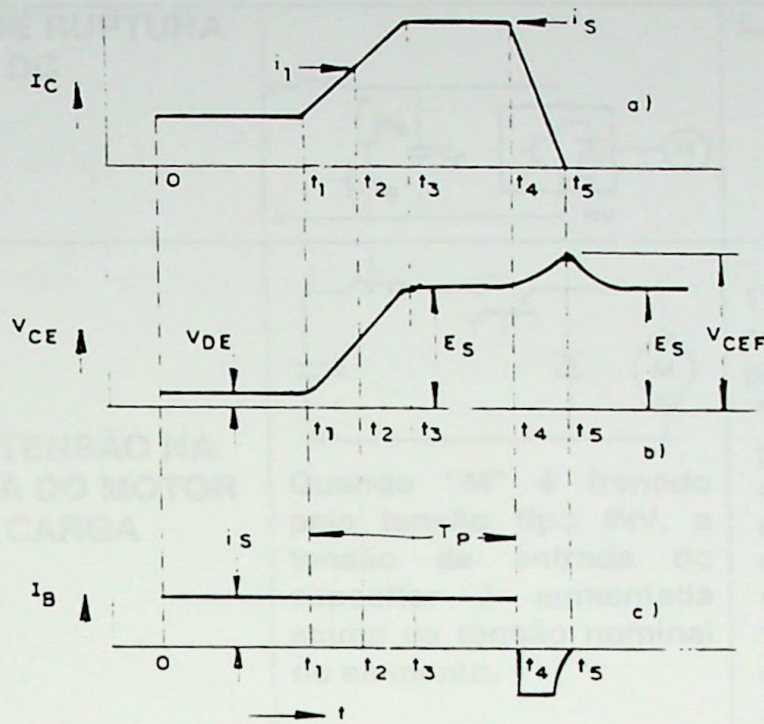
PERCURSO DO CURTO	CAUSA
<p>Curto no braço</p> 	<p>Destruição do transistor ou diodo</p>
<p>Curto no braço serial</p> 	<p>Falha do circuito de controle ou circuito de acionamento ou mau funcionamento devido a ruído</p>
<p>Curto na saída</p> 	<p>Erro do operador, tal como erro de fiação ou ruptura dielétrica</p>
<p>Curto da carga ao terra</p> 	<p>Erro do operador, tal como erro de fiação ou ruptura dielétrica</p>

Tabela 2 - Causas de sobrecorrente num Inversor

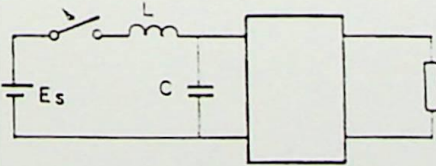
A figura 26 a seguir, mostra o comportamento dos principais parâmetros DC do transistor, em especial o intervalo  $t_4 - t_5$  onde está retratado a ação de detecção e bloqueio da sobrecorrente.



- $I_1$  - Nível de sobrecorrente estabelecido anteriormente
- $I_S$  - Corrente máxima de curto-circuito
- $E_S$  - Tensão da fonte
- $V_{CEF}$  - Tensão máxima de ponta
- $T_V$  - Tempo de retardo na detecção de sobrecorrente por transistor

Figura 26 - Processo de Geração de Sobrecorrente de um Ramo

A tabela 3 mostra as causas que podem levar à ocorrência de sobretensão, e suas respectivas soluções.

ITEM	CAUSA E CARACTERÍSTICA	SOLUÇÃO
SURTO ESTÁTICO	<p>A sobretensão de curto tempo é aplicado à entrada do dispositivo por surto de chaveamento.</p> 	<ol style="list-style-type: none"> <li>1) Insira circuito de controle de sobretensão, tal como absorvedor de surto ou "z - trap", dentro do circuito de entrada.</li> <li>2) Use enrolamento de blindagem estática no transformador.</li> <li>3) Para diodo retificador, conecte o capacitor de alta freqüência à saída DC.</li> </ol>
SURTO DE RUPTURA DC	<p>A energia de <math>L_d</math> é desviada para <math>C_d</math> na ruptura da corrente DC, aumentando a tensão em <math>C_d</math>.</p>	<ol style="list-style-type: none"> <li>1) Use <math>C_d</math> como absorvedor de surto.</li> <li>2) Selecione <math>C_d</math> que case</li> </ol>

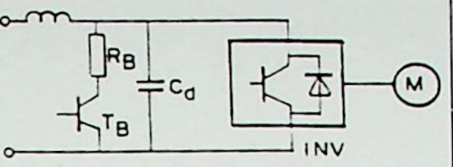
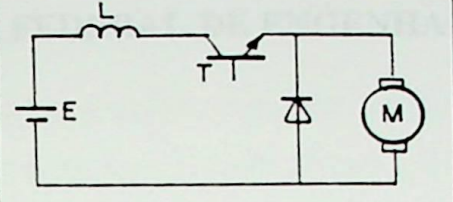
<p><b>SURTO DE RUPTURA DC</b></p>		<p><math>L_d</math> e a corrente DC.</p>
<p><b>SOBRETENSÃO NA RUPTURA DO MOTOR DE CARGA</b></p>	 <p>Quando "M" é frenado pela tensão tipo INV, a tensão de entrada do capacitor é aumentada acima da tensão nominal do elemento.</p>	<p>1) Cheque a tensão em <math>C_d</math> sem atraso e execute os passos seguintes:                  * use <math>R_b</math> para absorver a potência de frenagem e corte <math>T_B</math>, conforme necessário, durante controle da tensão variável.                  * remova o comando de controle INV, se necessário</p>
<p><b>TENSÃO DE PONTA DURANTE CHAVEAMENTO DO DISPOSITIVO DE POTÊNCIA</b></p>	<p>A tensão de ponta é desenvolvida pela indutância da linha de alimentação quando <math>T_w</math> está desligado.</p>	<p>Use "snubber" de controle de sobretensão e, selecione a constante adequadamente.</p>

Tabela 3 - Causas de Sobretensão e Características

O capítulo seguinte vai detalhar exemplos de circuitos eletrônicos que, tem a finalidade de recolher a informação do estado de um parâmetro, de forma que possa indicar falha da UPS.



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

### CAPÍTULO 3

## HARDWARE PARA DETECÇÃO DE DEFEITOS E SUPERVISÃO

## Capítulo 3 - HARDWARE PARA DETECÇÃO DE DEFEITOS E SUPERVISÃO

### 3.1-) Considerações Iniciais

O presente capítulo pretende mostrar circuitos eletrônicos que são utilizados para monitorar parâmetros relevantes no tocante à integridade de circuitos de potência.

Para melhor entendimento, dividiu-se o conteúdo em três tópicos:

- Falhas do Sistema: pertinentes a falhas que não ocorrem no circuito de potência;
- Pontes Trifásicas Tiristorizadas: circuitos voltados para monitoramento de uma ponte trifásica totalmente controlada;
- Pontes Inversoras: circuitos que acompanham o funcionamento adequado de pontes inversoras do tipo Fonte de Tensão.

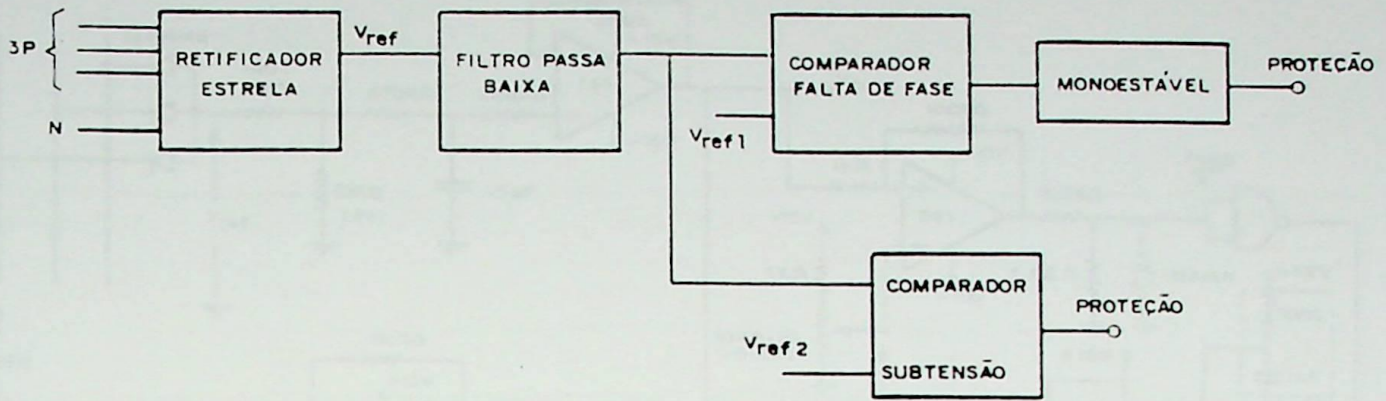
Alguns desses circuitos foram simulados em computador, cujos resultados obtidos estão comentados no item 3.5.

### 3.2-) Hardware para Falhas do Sistema

Os primeiros circuitos a serem mostrados [2], são relativos ao sinal trifásico, que alimentam o circuito de potência.

#### a) Circuito 1: Falta de Fase e Subtensão

O diagrama em blocos, mostrado na figura 27, fornece uma idéia geral do circuito utilizado para detecção de falta de fase e subtensão.



**Figura 27 – Falta de Fase e Subtensão**

O circuito funciona da seguinte forma:

- Inicialmente, retifica-se o sinal  $3\phi$  e, através do filtro passa-baixa ( $f_c \cong 2 \text{ Hz}$ ), obtém-se o valor médio do sinal retificado. Este valor é de aproximadamente, 60% do valor RMS do sinal  $V_{REF}$  em condições normais.
- O valor médio encontrado é aplicado em dois comparadores de histerese (para melhorar a imunidade do circuito a ruídos) e, comparado com dois valores de referência ( $V_{REF1}$  e  $V_{REF2}$ ). Em condições normais, o valor médio do sinal retificado é maior que os dois valores e, os comparadores estarão em um dado nível. Por outro lado se houver uma subtensão ou uma falta de fase no sistema, o valor médio da tensão retificada será agora menor que uma das  $V_{REF}$  (dependendo da anormalidade ocorrida no sistema  $3\phi$ ). Logo, um dos comparadores mudará de nível, o que atuará na proteção do sistema de disparo do circuito de potência.

OBS: Na saída do comparador de falta de fase, há ainda um monoestável ( $t_P = 30 \text{ s}$ ), o mesmo serve para proporcionar um tempo de espera no religamento do circuito, caso a falta de fase existente, deixe de existir (segurança).

O circuito final encontrado, é mostrado na figura 28.

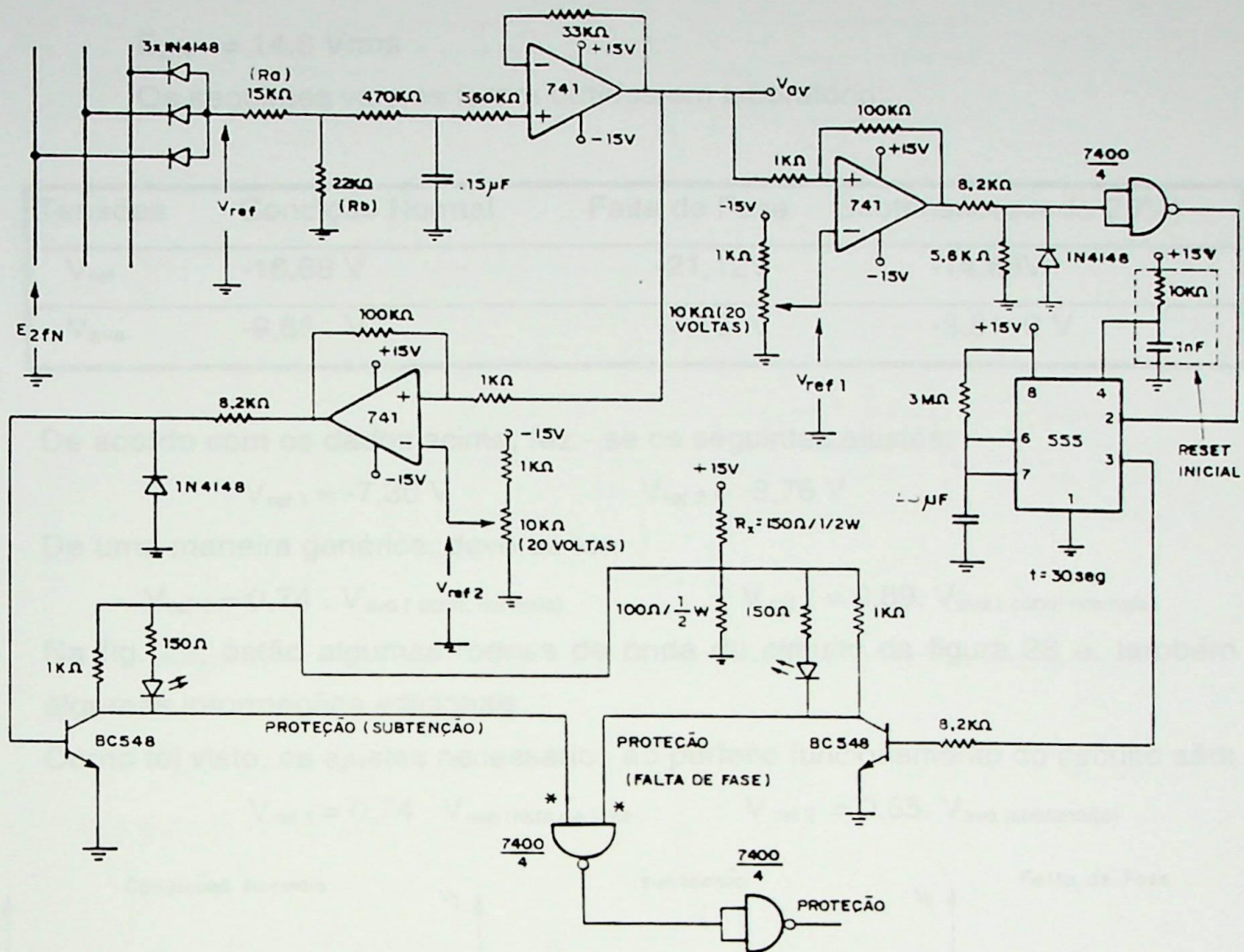


Figura 28 - Circuito para Detecção de Falta de Fase e Subtensão

A rede atenuadora formada pelos resistores  $R_a = 15\text{ K}\Omega$  e  $R_b = 22\text{ K}\Omega$ , serve para evitar a saturação do 1º amplificador operacional e, com os valores adotados, o secundário do transformador de sincronismo fica limitado, no máximo  $20\text{ V}_{rms}$  (fase - neutro), pois caso contrário o Amp. Op. iria saturar e comprometer o funcionamento do circuito.

Para um melhor entendimento do circuito da figura 26, sejam as seguintes tensões:

- $E_{2fN}$  - tensão fase - neutro no secundário do transformador de sincronismo rms
- $V_{ref}$  - tensão 3Ø retificadora (valor médio)
- $V_{ave}$  - tensão proporcional valor médio de  $V_{ref}$  (devido à rede  $R_a$  e  $R_b$ )
- $V_{ref1}$  - tensão de referência para disparo do circuito (falta de fase)
- $V_{ref2}$  - tensão de referência para disparo do circuito (subtensão)

$E_{2fN} = 14,8 \text{ Vrms}$

Os seguintes valores foram obtidos em laboratório:

Tensões	Condição Normal	Falta de Fase	Subtensão(queda 25%)
$V_{ref}$	-16,68 V	-21,12V	-14,88V
$V_{ave}$	-9,86 V	-7,18 V	-8,81 0 V

De acordo com os dados acima, fez - se os seguintes ajustes:

$V_{ref 1} = -7,30 \text{ V}$

$V_{ref 2} = -8,76 \text{ V}$

De uma maneira genérica, deve-se ter:

$V_{ref 1} = 0,74 \cdot V_{ave} \text{ (cond. normais)}$

$V_{ref 2} = 0,89 \cdot V_{ave} \text{ (cond. normais)}$

Na fig. 29, estão algumas formas de onda do circuito da figura 28 e, também algumas informações adicionais.

Como foi visto, os ajustes necessários ao perfeito funcionamento do circuito são:

$V_{ref 1} = 0,74 \cdot V_{ave} \text{ (falta de fase)}$

$V_{ref 2} = 0,85 \cdot V_{ave} \text{ (subtensão)}$

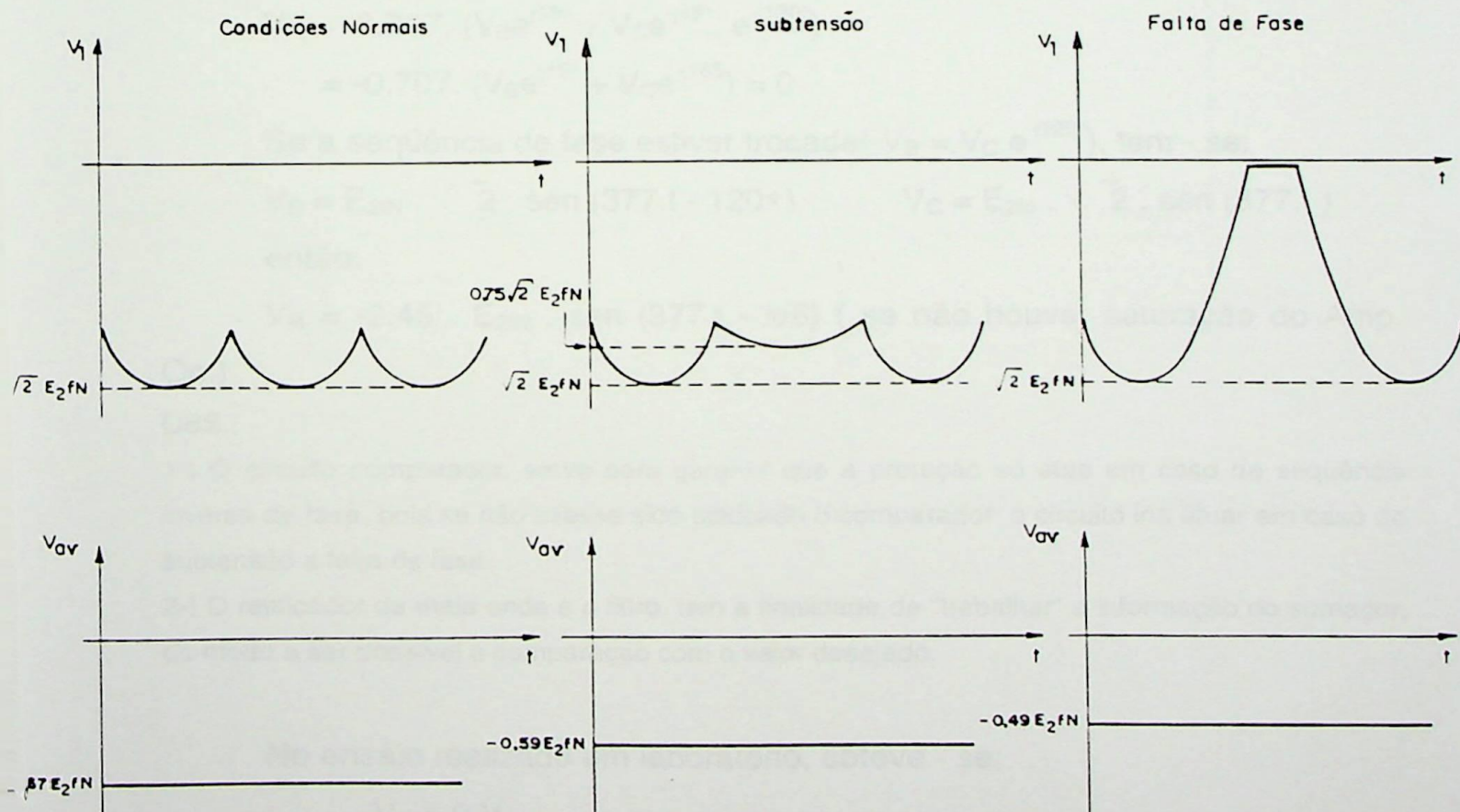


Figura 29 - Formas de Onda Típicas do Circuito da Figura 28

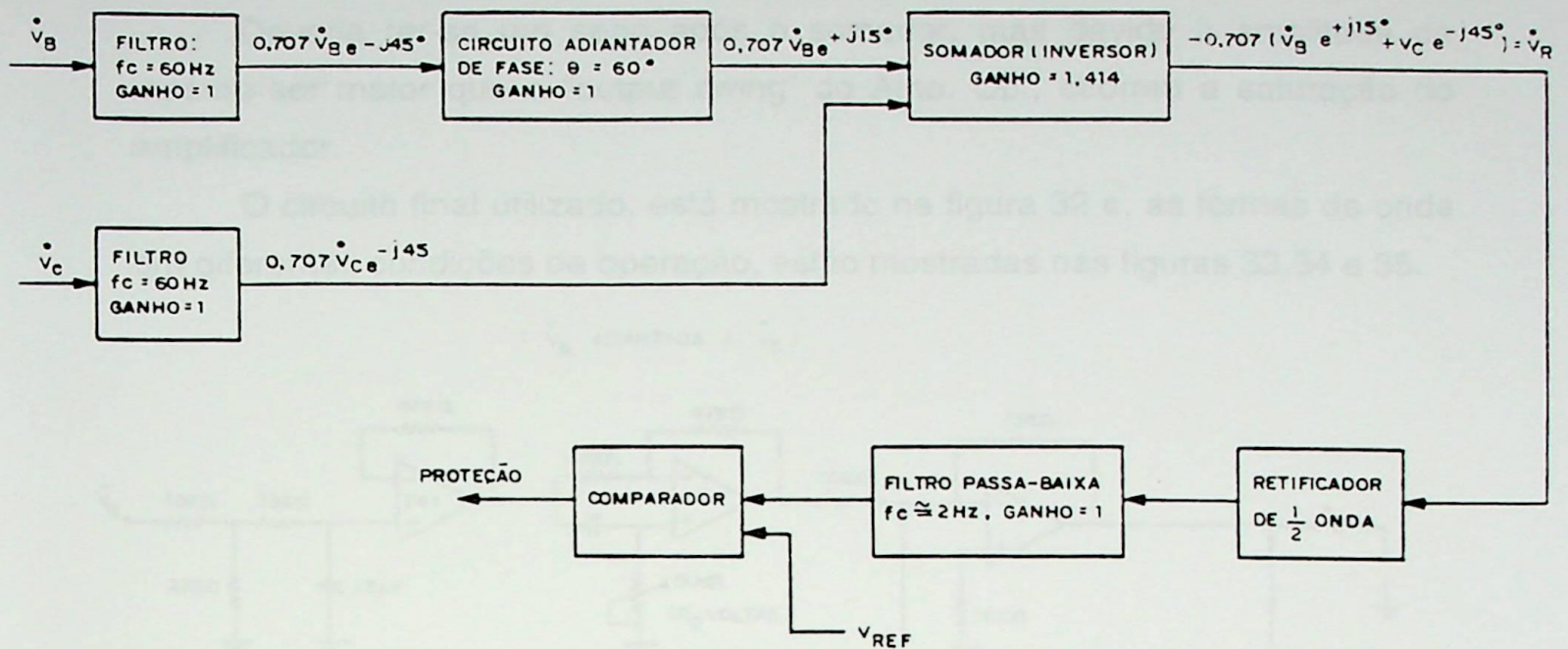


Figura 31 - Diagrama em Blocos de Seqüência Correta das Fases

Como foi visto,  $V_R = -0,707 \cdot (V_B e^{j15^\circ} + V_C e^{-j45^\circ})$ . Logo, se a seqüência de fase estiver correta ( $V_C = V_B e^{-j120^\circ}$ ), tem - se:

$$V_R = -0,707 \cdot (V_B e^{j15^\circ} + V_C e^{-j45^\circ} \cdot e^{-j120^\circ})$$

$$= -0,707 \cdot (V_B e^{j15^\circ} + V_C e^{-j165^\circ}) = 0$$

Se a seqüência de fase estiver trocada ( $V_B = V_C \cdot e^{j120^\circ}$ ), tem - se:

$$V_B = E_{2fN} \cdot \sqrt{2} \cdot \text{sen}(377 \cdot t - 120^\circ) \quad V_C = E_{2fN} \cdot \sqrt{2} \cdot \text{sen}(377 \cdot t)$$

então:

$$V_R = -2,45 \cdot E_{2fN} \cdot \text{sen}(377 \cdot t - \pi/6) \quad (\text{se não houver saturação do Amp.})$$

Op.)

OBS.:

1-) O circuito comparador, serve para garantir que a proteção só atue em caso de seqüência inversa de fase. pois se não tivesse sido colocado o comparador, o circuito iria atuar em caso de subtensão e falta de fase.

2-) O retificador de meia onda e o filtro, tem a finalidade de "trabalhar" a informação do somador, de modo a ser possível a comparação com o valor desejado.

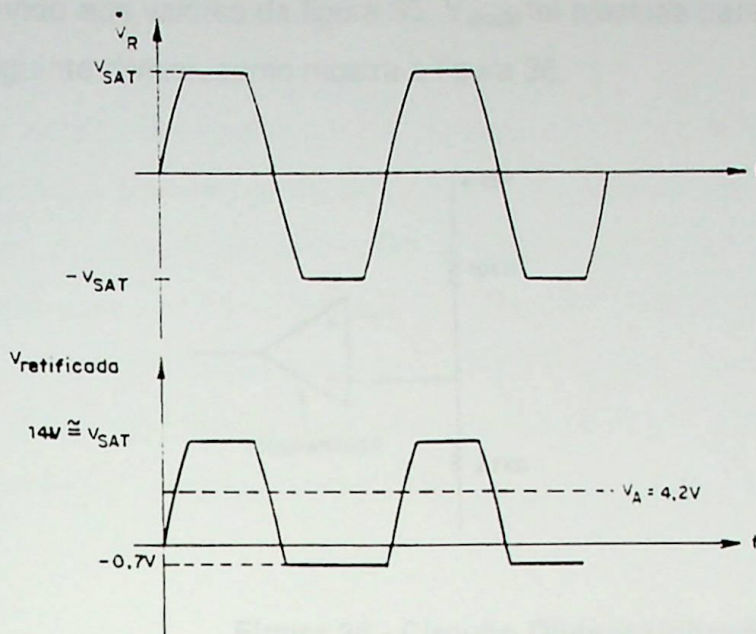
No ensaio realizado em laboratório, obteve - se:

$$V_R \approx 0 \text{ V}$$

Em condição de seqüência inversa de fase,

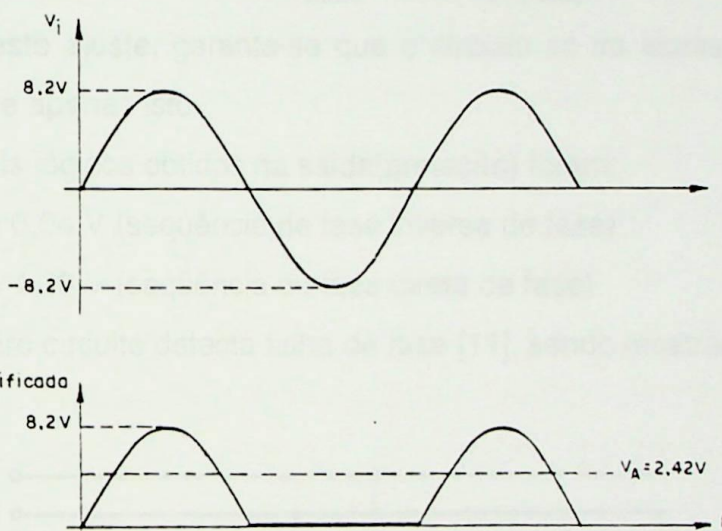
$$V_R = -34,3 \cdot \text{sen}(337 \cdot t - \pi/6) \text{ V.}$$





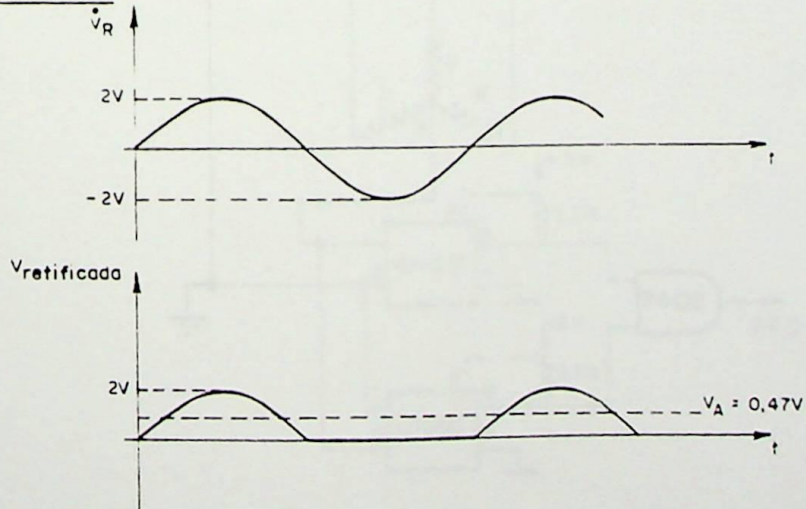
**Figura 33 - Formas de Onda na Inversão de Fase**

FALTA DE FASE



**Figura 34 - Formas de Onda na Falta de Fase**

SUBTENSÃO



**Figura 35 - Formas de Onda na Subtensão**

Devido aos valores da figura 35.  $V_{ajuste}$  foi ajustada para 3,8 V, ou seja, foi feito o seguinte divisor, como mostra a figura 36.

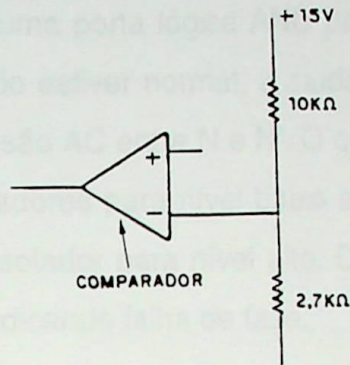


Figura 36 - Circuito Divisor Utilizado

De uma forma genérica,

$$V_{ajuste} = 0,22 \cdot E_{2IN(RMS)}$$

Como este ajuste, garante-se que o circuito só irá acusar seqüência de fase trocada, e apenas isto.

Os níveis lógicos obtidos na saída (proteção) foram:

Nível<sub>0</sub> = 0,04 V (seqüência de fase inversa de fase)

Nível<sub>1</sub> = 4,30 V (seqüência de fase direta de fase)

O terceiro circuito detecta falha de fase [11], sendo mostrado na figura 37.

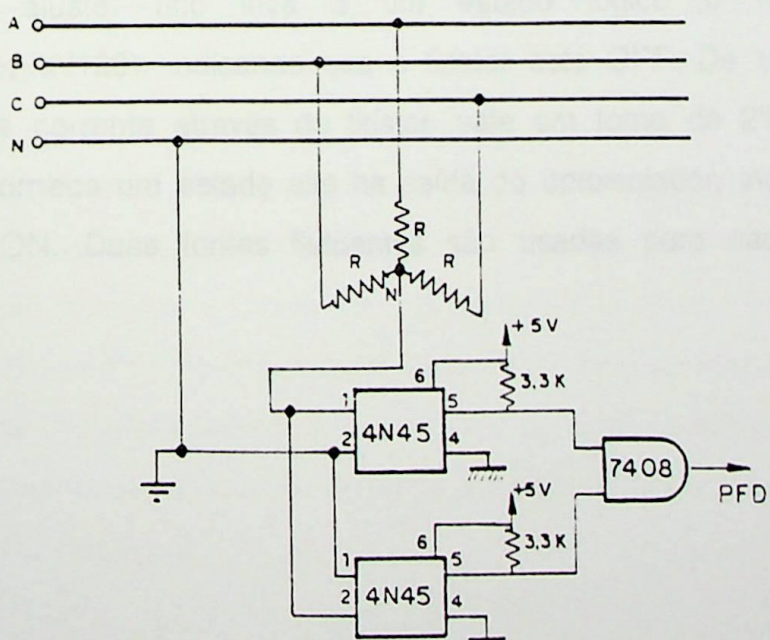


Figura 37 - Circuito de Detecção de Falha de Fase

Na presença de todas as fases, a tensão entre N e N' é zero (admitindo a fonte de alimentação balanceada).

Isto produz um estado alto na saída de ambos os optoisoladores, os quais são conectados por uma porta lógica AND para produzir um sinal que reflita tal estado, isto é, se tudo estiver normal, a saída é ON. Ausência de qualquer das fases causa uma tensão AC entre N e N'. O quarto positivo de ciclo, leva a saída de um dos optoisoladores para nível baixo e o quarto negativo de ciclo, leva a saída do outro optoisolador para nível alto. Deste modo, a saída da porta AND sempre será OFF, indicando falha de fase.

### 3.3-) Hardware para Pontes Trifásicas Tiristorizadas

Serão comentados dois circuitos, utilizados em [11], para detecção do estado dos SCRs, e da corrente de saída da ponte.

O primeiro circuito, mostrado na figura 38, detecta condição "on/off" de um determinado SCR, além da ocorrência de curto-circuito sobre o mesmo.

Um resistor "shunt" é ligado em série com o componente de potência. A tensão através do shunt determina o sinal de corrente que flua em ambos os dispositivos. Esta tensão é levada a um comparador. Na ausência de corrente sobre o SCR, o comparador é ajustado para alcançar saturação negativa, via off-set de ajuste. Isto leva a um estado lógico 0 na saída do optoisolador (4N139), indicando que o tiristor está OFF. De outra forma, a presença de corrente através do tiristor, vale em torno de 2% da corrente circulante, fornece um estado alto na saída do optoisolador, indicando que o SCR está ON. Duas fontes flutuantes são usadas para cada circuito de detecção.

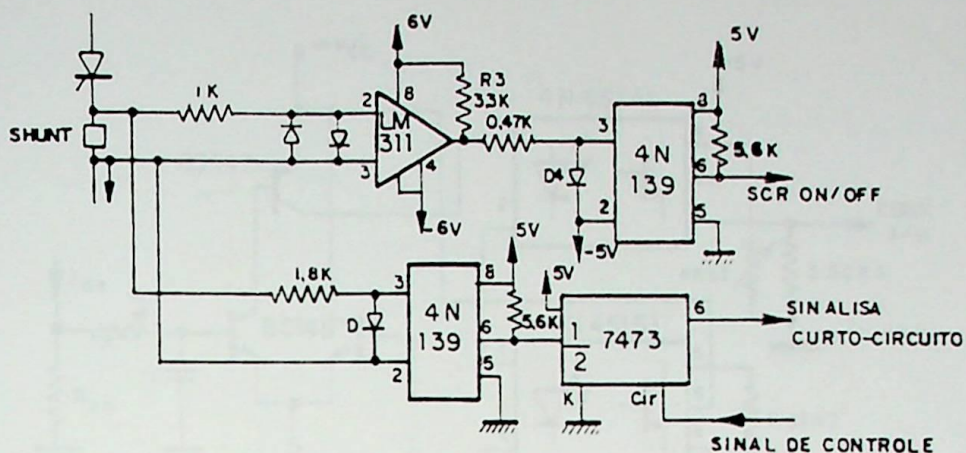


Figura 38 - Circuito de Detecção de ON/OFF e Curto-Circuito

Sob condições normais de operação, a corrente através do tiristor é nula ou positiva. Isto leva a uma tensão positiva ou nula através do "shunt", levando a saída do 2º optoisolador para nível alto. Supondo que um SCR(T3 por exemplo) é curto - circuitado e outro tiristor do mesmo grupo(T5) é disparado, uma grande corrente reversa flua através de T3. Esta corrente reversa desenvolve uma grande tensão negativa sobre o shunt, o qual leva a saída do correspondente optoisolador para nível baixo. Um "latch"(7473) é usado para manter esta informação.

O outro circuito, mostrado na figura 39, representa um laço de realimentação da corrente DC  $I_{DC}$ .

A corrente é obtida, usando um resistor tipo shunt(em torno de  $100 \Omega$ , em série com a carga. O filtro R-C é do tipo passa-baixa, com constante de tempo  $T_F$  ajustável.

O sinal filtrado é então opticamente isolado. Este estágio de isolamento compreende um amplificador diferencial e dois optoisoladores(4N45). Uma fonte de alimentação flutuante é usada para alimentar tanto o amplificador como o optoisolador B, sendo diferente a alimentação para o optoisolador A(nível TTL).

A saída do optoisolador A é proporcional ao sinal da corrente. Este sinal analógico é enviado a um sistema digital de supervisão, utilizando conversor A/D e uma porta I/O 8212.

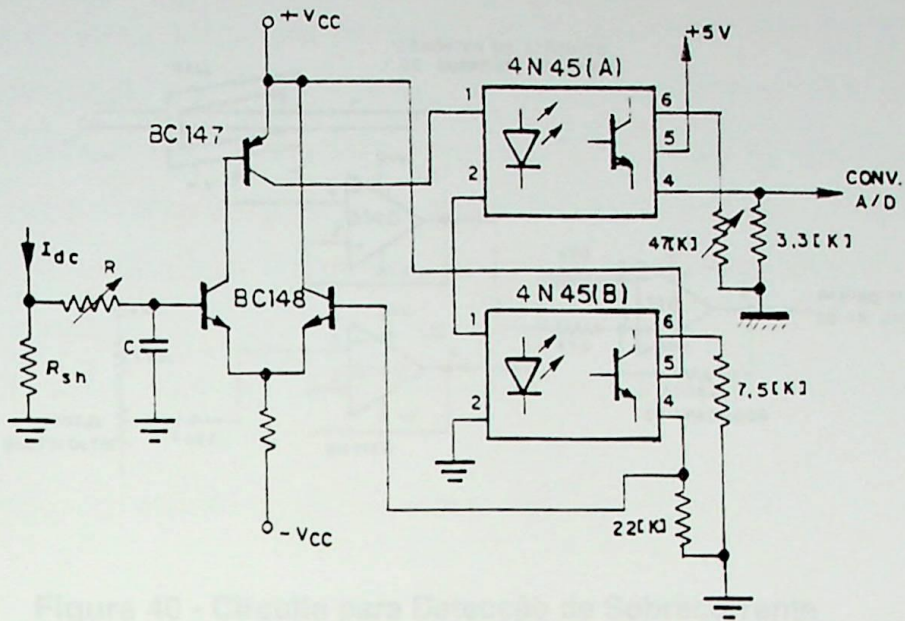


Figura 39 - Circuito de Detecção da Corrente de Saída DC

### 3.4-) Hardware para Inversores Tipo Fonte de Tensão

Este item apresenta dois circuitos específicos para componente eletrônico de potência mais utilizado em pontes inversoras, o IGBT(transistores de gate isolado).

O circuito da figura 40, ilustra a detecção de sobre-correntes que pode ocorrer devido a curto circuitos provocados pela condução simultânea entre chaves dos pólos positivo e negativo do inversor de frequência ou devido a problemas na geração dos sinais de gatilhamento dos IGBT's.

A supervisão de sobrecorrente é feita por um sensor Hall, com capacidade de medida para 50 A. O sinal de corrente é aplicado a um circuito isolador e em seguida a entrada inversora do circuito comparador, implementado com circuito integrado linear LM318.

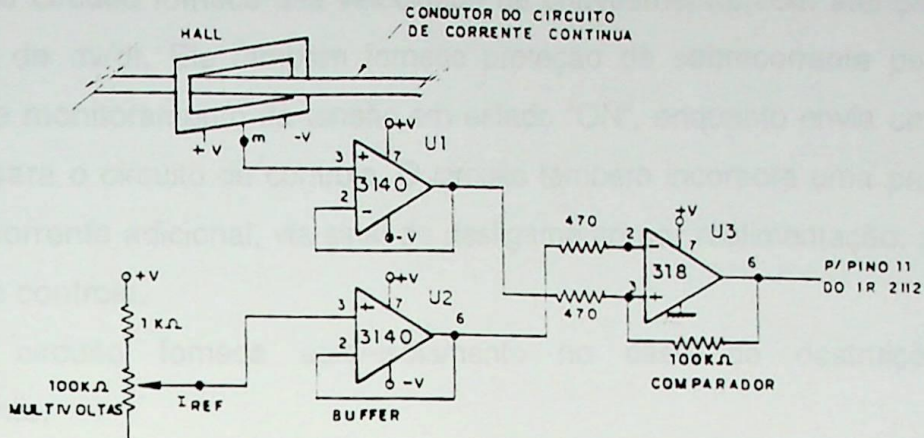


Figura 40 - Circuito para Detecção de Sobrecorrente

A comparação feita entre os sinais obtidos do sensor Hall e o de referência, proveniente de outro buffer, leva a saída do comparador a saturação positiva ou ao valor zero. Esta saída é aplicada ao pino 11 do circuito pré - amplificador ou drive IR2112[13]. Na ocorrência de qualquer sobrecorrente, a saturação positiva do comparador inibe imediatamente os pulsos de gatilhamento das chaves que formam o circuito que opera em condição de curto - circuito; ou seja, por onde circularia a sobrecorrente.

A figura 41, ilustra a ação de inibição dos sinais de gatilhamento dos IGBT's.

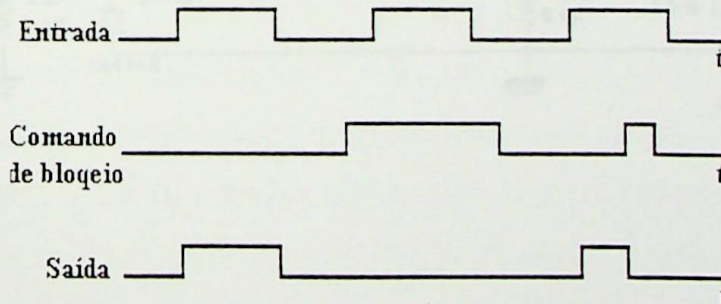


Figura 41 - Sinais de Entrada /Saída do Pré - Amplificador e o Comando de Bloqueio por Sobrecorrente

É apresentado a seguir [13] um circuito drive específico para transistores tipo IGBT.

Este circuito fornece alta velocidade de chaveamento, com atenção aos requisitos de dv/dt. Ele também fornece proteção de sobrecorrente pulsada, através de monitoramento da tensão em estado "ON", enquanto envia um sinal de falha para o circuito de controle. O circuito também incorpora uma proteção de sobrecorrente adicional, via sinal de desligamento por realimentação, para o circuito de controle.

O circuito fornece auto-isolamento no caso de destruição do componente.

O esquema elétrico do circuito está mostrado na figura 42.

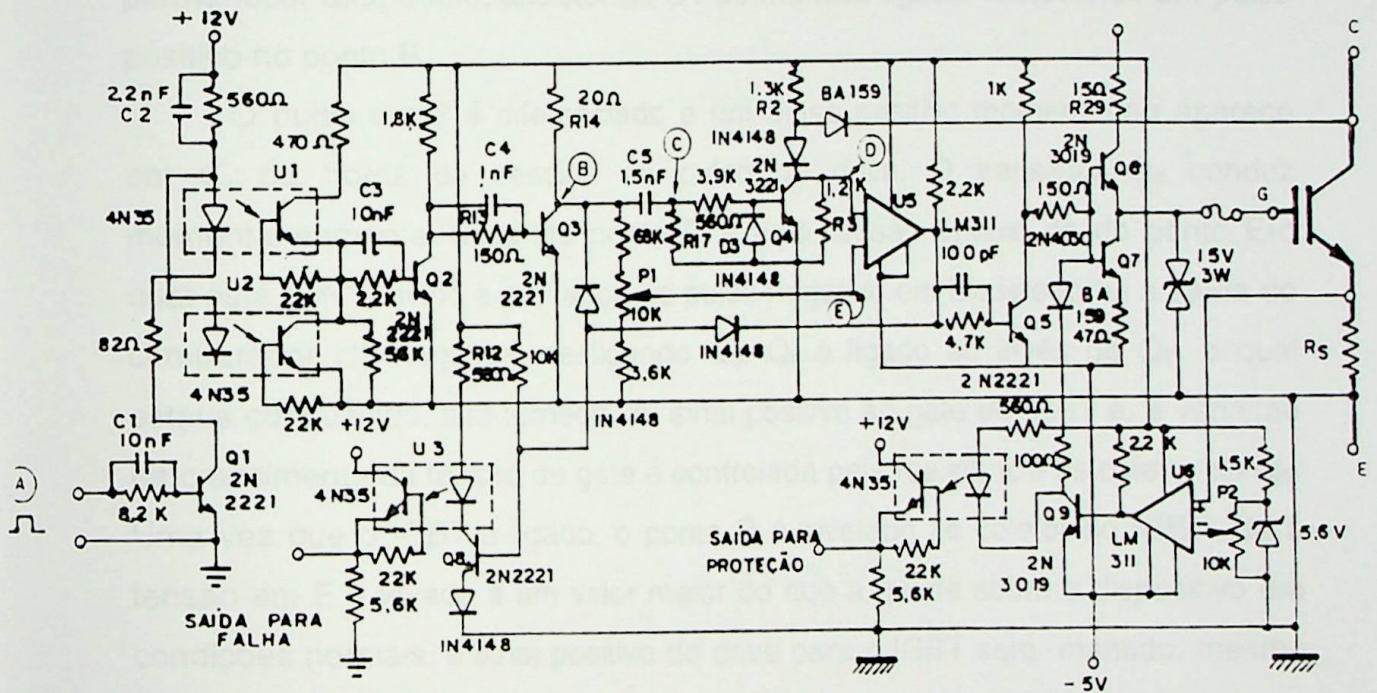


Figura 42 – Circuito de Drive para IGBT

Quando o sinal de entrada em A é baixo, a entrada invertida E do comparador U1 está em potencial zero, enquanto a entrada não - invertida D possui tensão dada por:

$$V_D = R3 \cdot (V_S - V_{D2}) / (R2 + R3)$$

onde  $V_S$  é a fonte positiva de alimentação, +15 V

$V_{D2}$  é a queda de tensão através do diodo D2

Deste modo, a saída do comparador é positiva, mantendo  $Q_5$  e  $Q_7$  conduzindo, de modo que a porta do IGBT fica reversamente polarizada, pela fonte de tensão negativa (-5 V), mantendo o dispositivo desligado.

Os pulsos de controle em A para o IGBT são gerados por um controle lógico, levando o transistor  $Q_7$  para condução, permitindo o fluxo de corrente sobre o LED do optoisolador U1. Na borda de crescimento do pulso, uma parte da resistência série do LED é curto-circuitada através de um pequeno capacitor, tal que o pico de corrente resultante causa uma condução mais rápida do LED, reduzindo o tempo de condução. Se o sinal em A permanecer alto, o fototransistor de U1 permanece ligado, fornecendo um pulso positivo no ponto B.

O pulso em B é diferenciado e um pulso positivo momentâneo aparece em C, na borda de descida do pulso do drive. O transistor  $Q_4$  conduz momentaneamente, trazendo ponto D a uma tensão abaixo da do ponto E (o qual está "ON" devido à presença do pulso negativo em B). Isto torna a saída do comparador U5 negativa, desligando  $Q_5$ .  $Q_6$  é ligado ao invés de  $Q_7$ , o qual estava conduzindo. Isto fornece um sinal positivo ao gate do IGBT e, a variação de crescimento da tensão de gate é controlada pela resistência de coletor de  $Q_6$ . Uma vez que o IGBT é ligado, o ponto D é nivelado ao coletor do IGBT. Se a tensão em E é levada a um valor maior do que a queda sobre o dispositivo em condições normais, o sinal positivo do drive para o IGBT será mantido, mesmo se o pulso em C desaparecer após um período curto de tempo.

Se o sinal de entrada em A é trazido para zero pelo corte do IGBT, o sinal desaparece do ponto B, levando a tensão em E para um valor abaixo do valor do ponto D. Isto leva a saída do comparador U5 para uma tensão positiva, substituindo a tensão positiva no drive por um valor negativo, desligando o IGBT.

A variação de carga reversa no gate é controlada pelo resistor no coletor de  $Q_7$ .

Durante o período de conversão do IGBT, a tensão em D acompanha a queda de tensão sobre o componente. Caso esta queda seja mais acentuada devido a uma sobrecorrente, em torno da voltagem selecionada em E, a saída

do comparador irá variar positivamente, desligando o IGBT através da aplicação de um sinal negativo no gate por parte do drive.

O valor da corrente de coletor o qual será sentida como sobrecarga pelo circuito, depende da queda progressiva do dispositivo e da tensão sobre o ponto E. Desta forma, o nível de interrupção da sobrecorrente pode ser ajustado, via potenciômetro P1, como mostrado na figura 42.

As formas de onda em alguns pontos importantes do circuito, são mostradas na figura 43, tanto para condições normais como sobrecorrente.

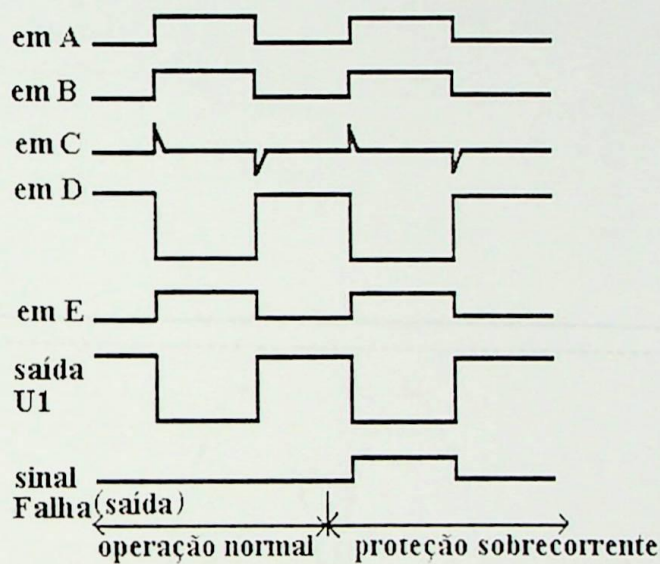
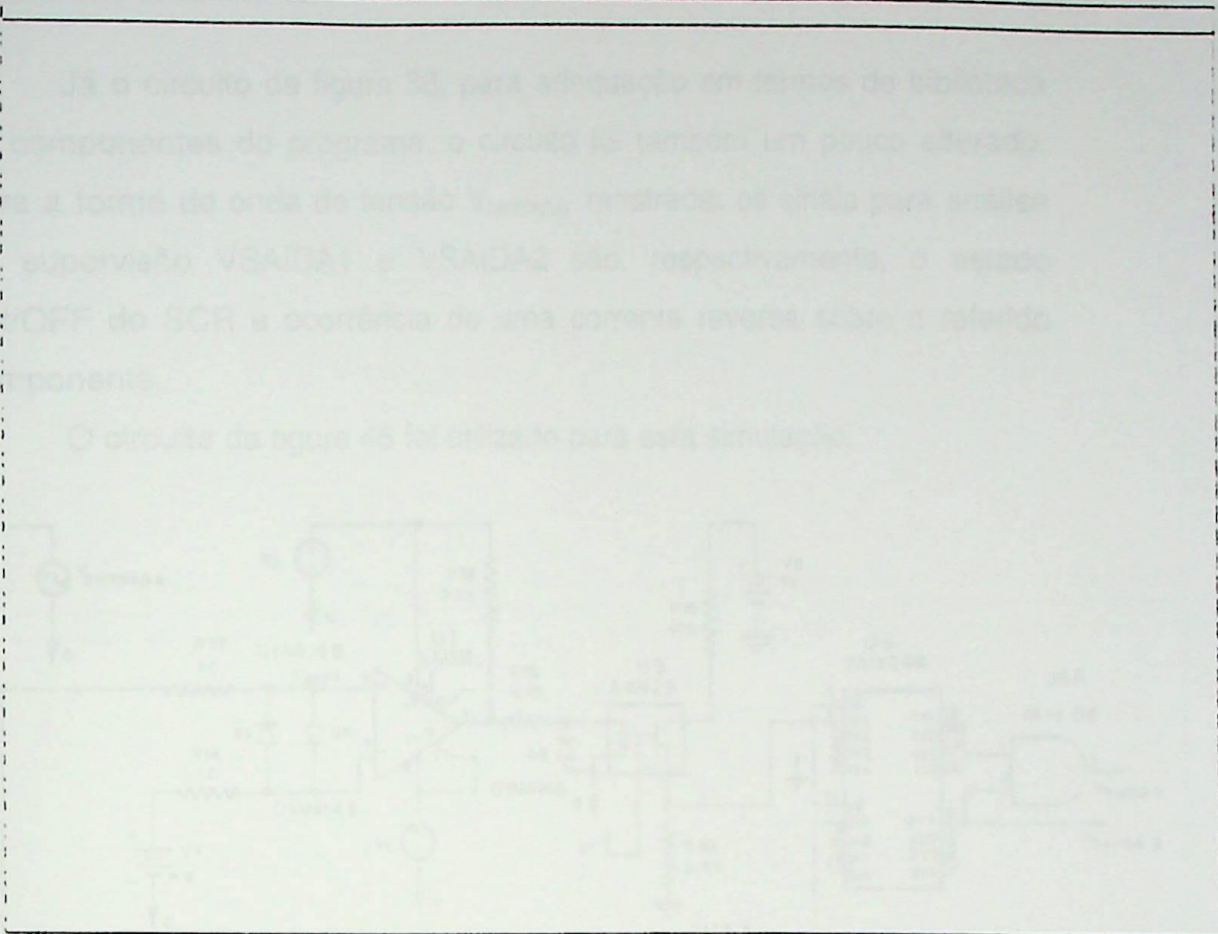


Figura 43 – Formas de Onda do Circuito de Drive em Operação

Desde que o dispositivo é levado à condução forçosamente na borda de crescimento do sinal de entrada por um pequeno pulso em C (o qual mantém a tensão em D abaixo do ponto E para duração do pulso), o IGBT não pode ser desligado durante este tempo, pela tensão de anodo, caso a corrente cresça acima do valor previsto. Deste modo, é melhor manter a largura do pulso em C tão curta quanto possível, tipicamente em torno de 10  $\mu$ s. enquanto uma pequena indutância em série com o coletor do IGBT, auxiliará em retardar a variação de aumento da corrente de falha durante este período “neutro”.

USADA



SEL>>

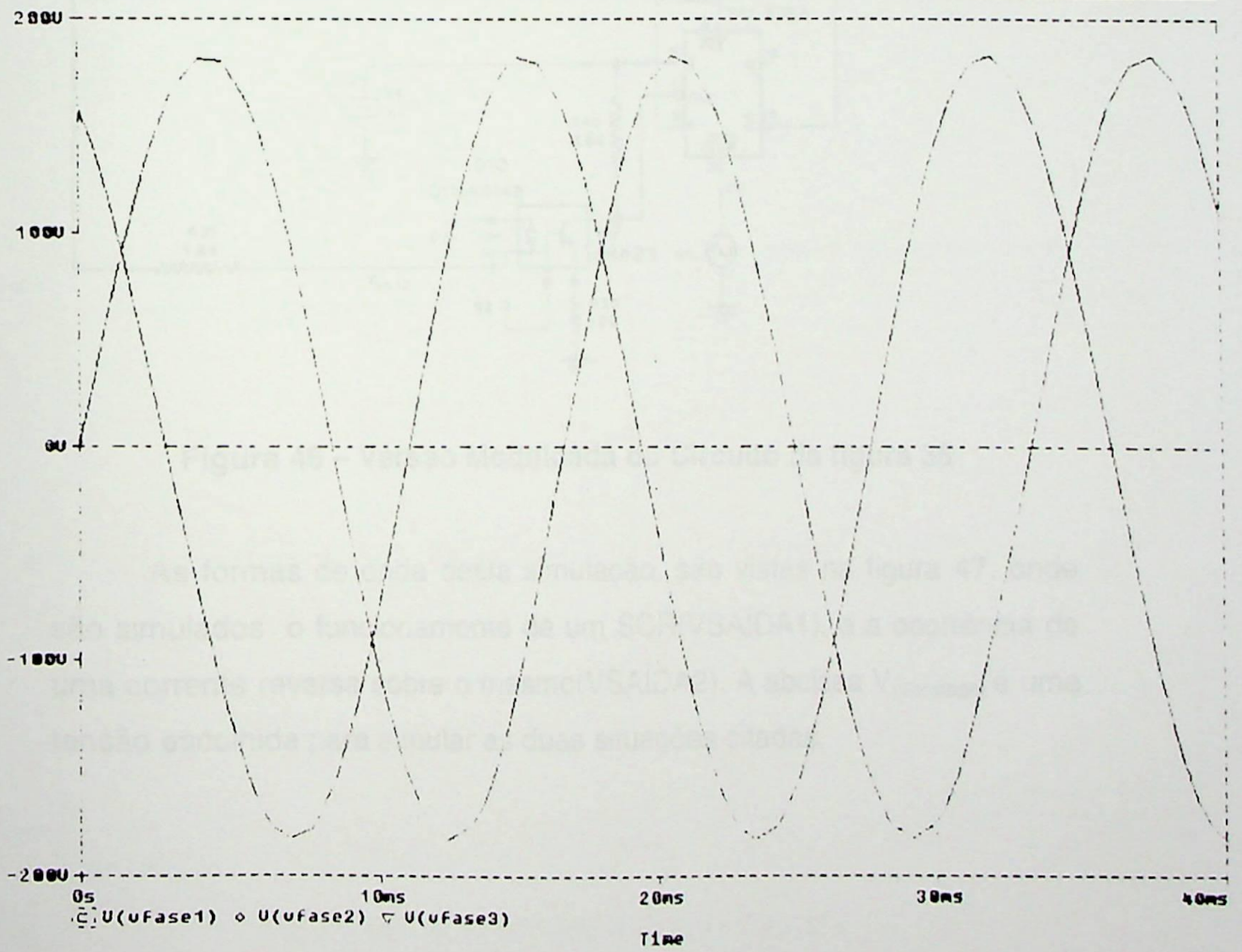


Figura 45- Falta da Fase 3

Já o circuito da figura 38, para adequação em termos de biblioteca de componentes do programa, o circuito foi também um pouco alterado. Para a forma de onda de tensão  $V_{(entrada)}$  mostrada, os sinais para análise da supervisão VSAIDA1 e VSAIDA2 são, respectivamente, o estado ON/OFF do SCR e ocorrência de uma corrente reversa sobre o referido componente.

O circuito da figura 46 foi utilizado para esta simulação.

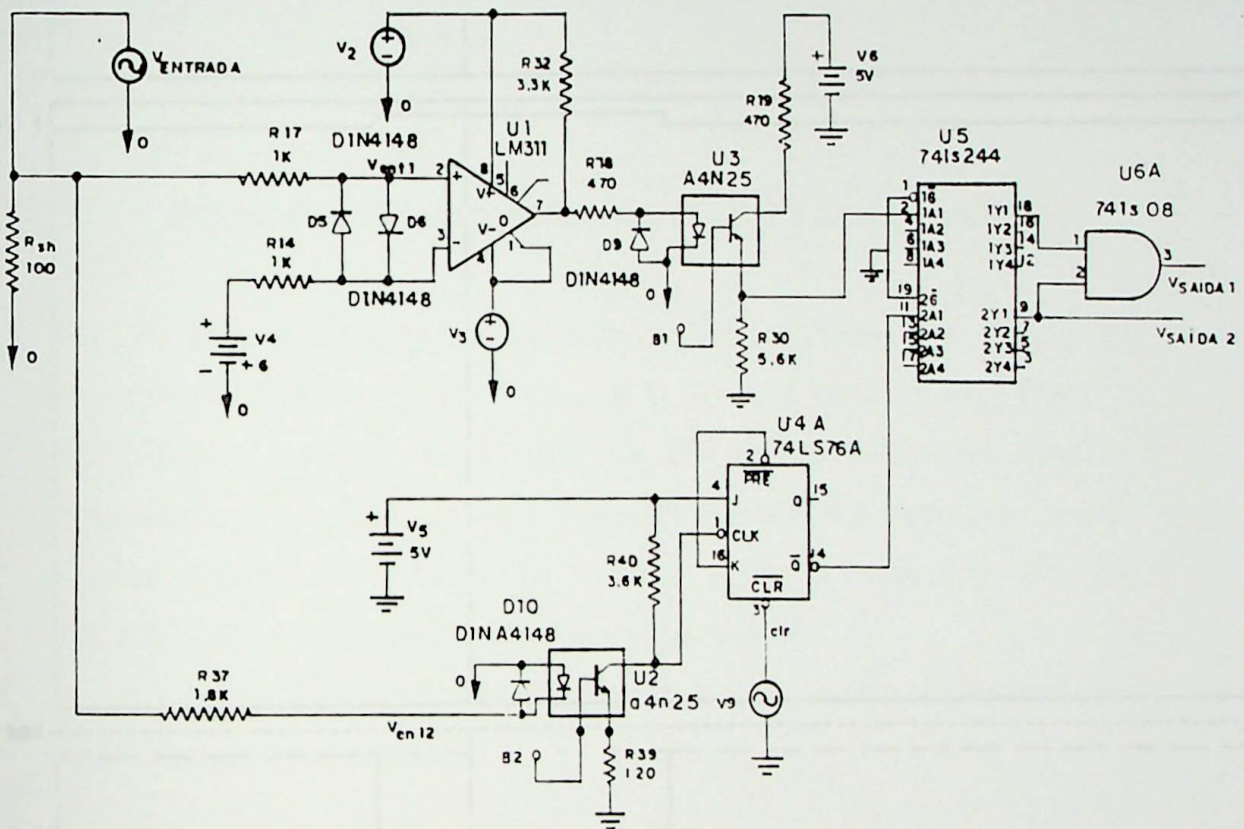


Figura 46 – Versão Modificada do Circuito da figura 38

As formas de onda desta simulação, são vistas na figura 47, onde são simulados o funcionamento de um SCR(VSAÍDA1), e a ocorrência de uma corrente reversa sobre o mesmo(VSAÍDA2). A abcissa  $V_{(entrada)}$  é uma tensão escolhida para simular as duas situações citadas.

Figura 47- SCR ON/OFF e Curto - Circuito

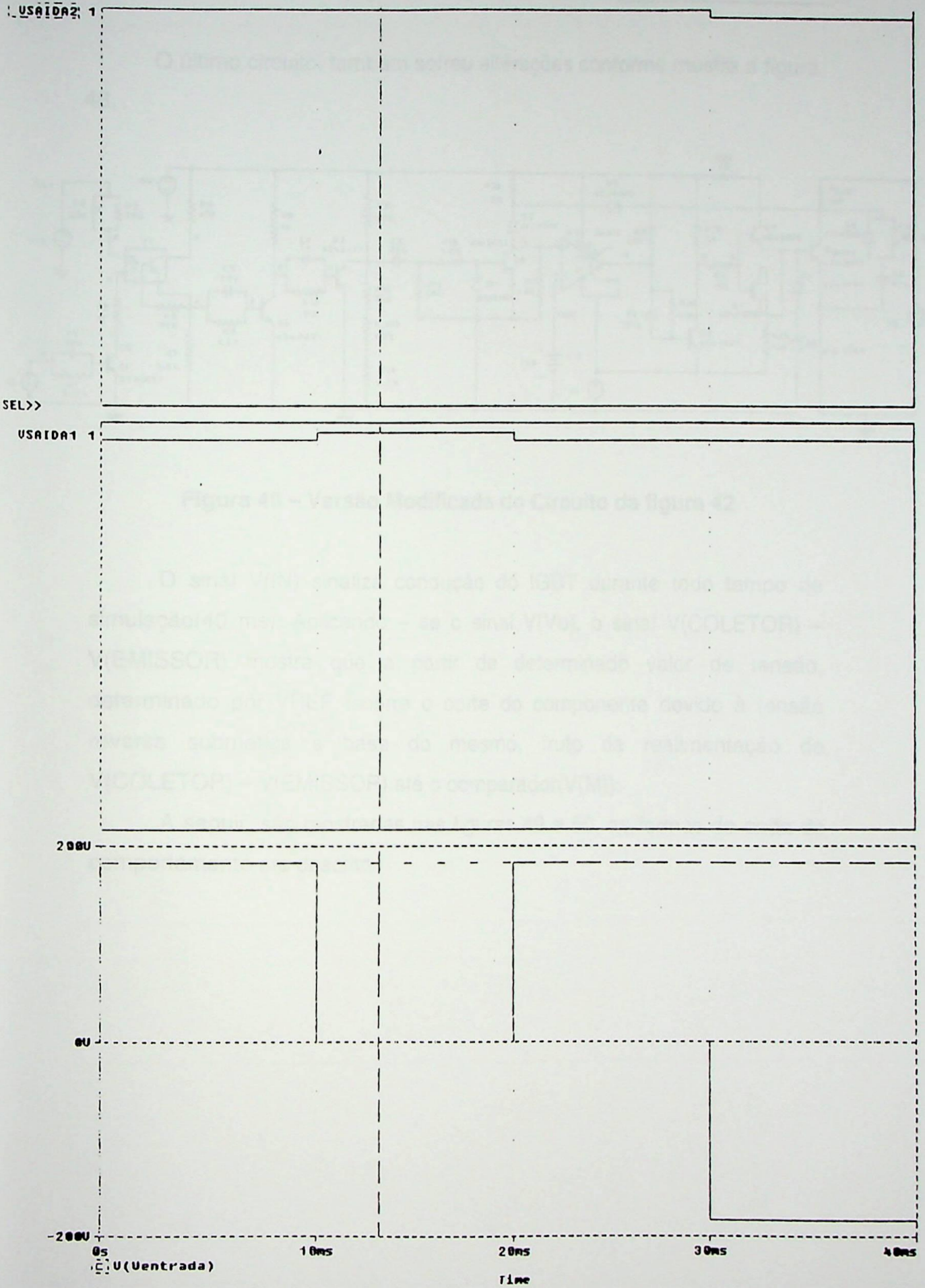


Figura 47- SCR ON/OFF e Curto - Circuito

O último circuito, também sofreu alterações conforme mostra a figura 48.

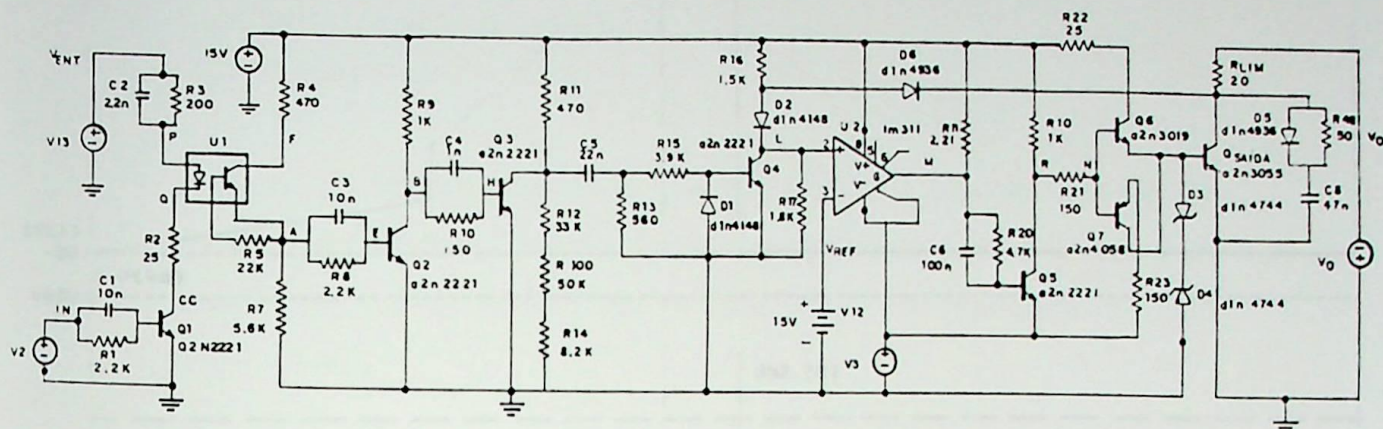


Figura 48 – Versão Modificada do Circuito da figura 42

O sinal V(IN) sinaliza condução do IGBT durante todo tempo de simulação(40 ms). Aplicando – se o sinal V(Vo), o sinal V(COLETOR) – V(EMISSOR) mostra que a partir de determinado valor de tensão, determinado por VREF, ocorre o corte do componente devido à tensão reversa submetida a base do mesmo, fruto da realimentação de V(COLETOR) – V(EMISSOR) até o comparador(V(M)).

A seguir, são mostradas nas figuras 49 e 50, as formas de onda do comportamento ora descrito.

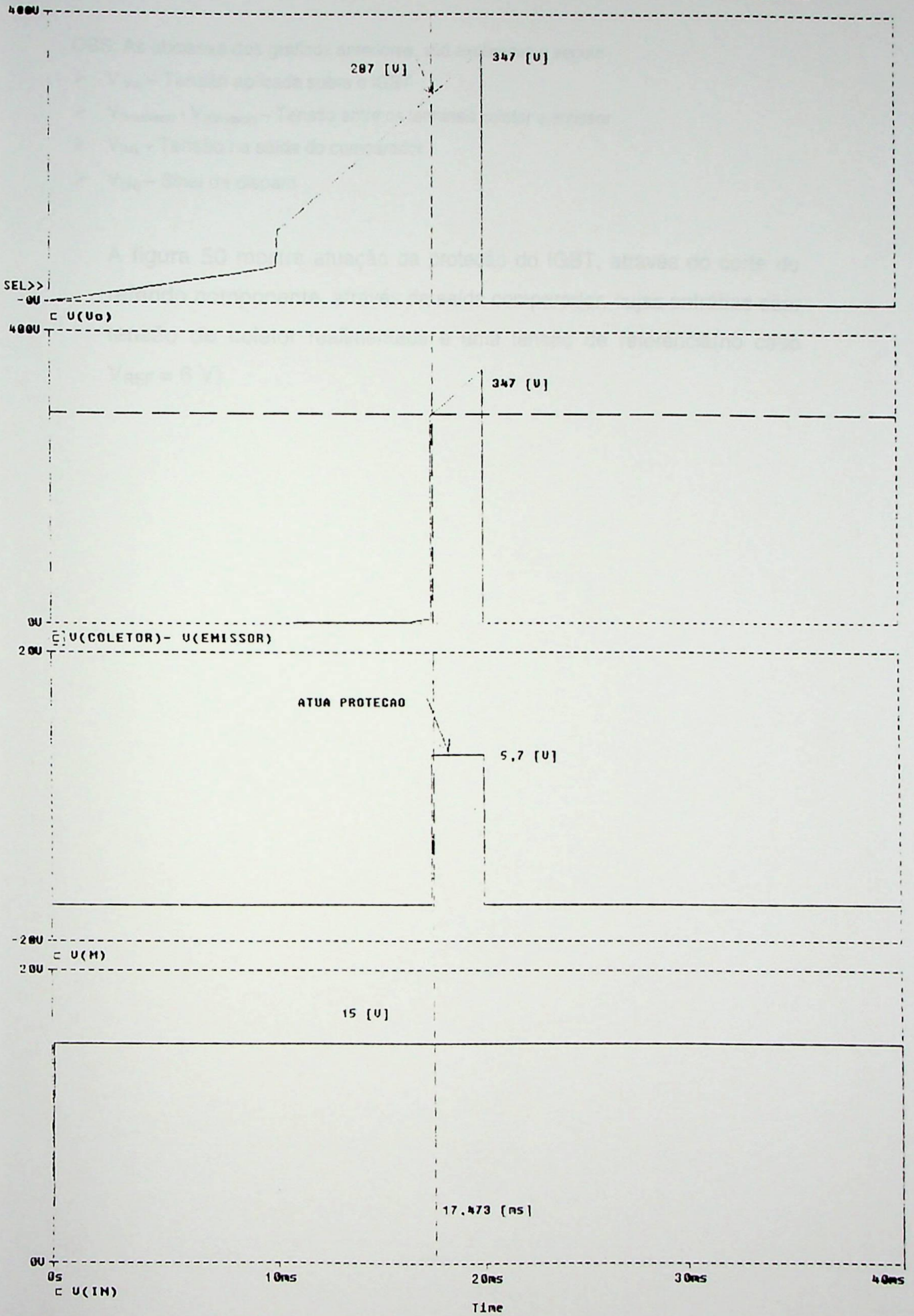


Figura 49- Formas de Onda para o Drive

OBS: As abcissas dos gráficos anteriores, são explicadas a seguir:

- $V_{(V_0)}$  – Tensão aplicada sobre o IGBT
- $V_{(V_{\text{coletor}})} - V_{(V_{\text{emissor}})}$  – Tensão entre os terminais coletor e emissor
- $V_{(M)}$  – Tensão na saída do comparador
- $V_{(IN)}$  – Sinal de disparo

A figura 50 mostra atuação da proteção do IGBT, através do corte do referido componente, através da saída comparador, cujas entradas são: tensão de coletor realimentada e uma tensão de referência (no caso  $V_{REF} = 6 \text{ V}$ ).

Figura 50 – Proteção Ativa sobre IGBT.

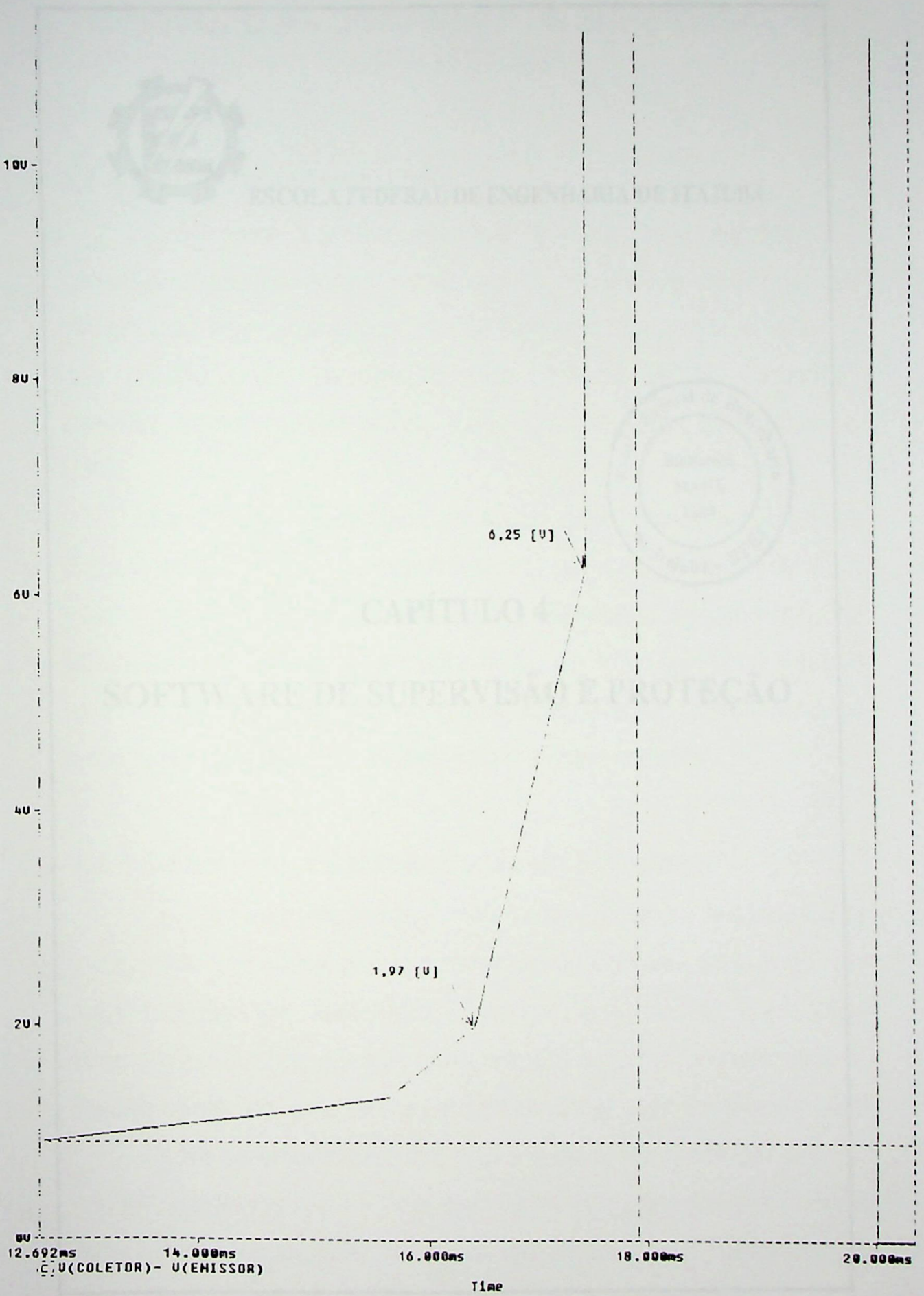


Figura 50- Proteção Atuou sobre IGBT



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ



## CAPÍTULO 4

# SOFTWARE DE SUPERVISÃO E PROTEÇÃO

### 4.2-) Descrição e Caracterização de Software

Com o crescente desenvolvimento da tecnologia, as respostas às excitações por parte dos componentes de potência são cada vez mais rápidas. Em vista disso, o software que interage com sinais vindos destes componentes, deve sempre ter um processamento consequente resposta rápida para acompanhar esse comportamento.

Outra característica que o software deve ter neste caso é que o conjunto de instruções deve permitir manipulação dos parâmetros com grande velocidade, isto é, ser um software com características

## CAPÍTULO 4 – SOFTWARE DE SUPERVISÃO E PROTEÇÃO

### 4.1-) Considerações Iniciais

O presente capítulo descreve a parte computacional da estratégia para monitoramento do circuito de potência em termos de integridade dos parâmetros: tensão, corrente, temperatura, não só dos componentes discretos (TBJs de potência, SCRs, Capacitor, Bateria), bem como dos blocos, vistos como um todo, que formam o UPS.

O software recebe sinais digitais e analógicos (convertidos em digitais pelos circuitos de monitoramento), compara cada um com um valor pré-definido. Em caso de discrepância (analógico) ou desigualdade (digital), o programa envia um sinal digital ao circuito externo, que irá tomar a decisão mais adequada para evitar dano ao circuito de potência e ao sistema que o mesmo controla.

### 4.2-) Descrição e Caracterização de Softwares

Com o crescente desenvolvimento da tecnologia, a resposta às excitações por parte dos componentes de potência são cada vez mais rápidas. Em vista disso, o software que interagir com sinais vindos destes componentes, deve sempre ter um processamento e, conseqüente resposta rápida para acompanhar esse comportamento.

Outra característica que o software deve ter, neste caso, é que o conjunto de instruções deve permitir manipulação dos parâmetros, com grande velocidade, isto é, ser um software com características industriais. Uma linguagem adequada a este ambiente é o "Assembly", que é o software usado pelos

microprocessadores(CPUs), componentes principais dos computadores; todavia, o software usado nestes componentes não possuem o que se chama de padronização, ou seja, um programa que é processado pelo microprocessador Z-80 da empresa ZILOG, não pode ser processado por um circuito baseado na CPU 6800 da MOTOROLA.

Pelo exposto acima, a utilização de programas que independam do hardware e também alcancem alta velocidade de processamento, podem ser utilizadas com sucesso na solução problemas em circuitos potência.

#### 4.3-) Perfil da Linguagem C

No presente trabalho foi utilizado uma linguagem de computador considerada médio nível, chamada linguagem C. A tabela 4 [13], esclarece o termo médio nível em termos de comparação entre outros softwares comerciais.

SOFTWARE	CLASSIFICAÇÃO	CARACTERÍSTICAS PRINCIPAIS
ASSEMBLY	BAIXO NÍVEL	Muitas Instruções(> 100); Não – Portátil; Todas Instruções Acessam Diretamente o Hardware; Linguagem Não – Estruturada; Altíssima Velocidade de Processamento.
C	MÉDIO NÍVEL	Poucas Instruções(< 50); Portátil; Certas Funções Acessam Diretamente o Hardware; Linguagem Estruturada; Alta Velocidade de Processamento.
BASIC, COBOL, PASCAL	ALTO NÍVEL	Muitas Instruções(> 100); Portátil; Nenhuma Instrução Acessa o Diretamente Hardware; Linguagem Não – Estruturada; Média Velocidade de Processamento.

Tabela 4 – Comparação entre Softwares

Analisando a tabela comparativa sob a ótica de supervisão/proteção de circuitos eletrônicos, é importante analisar algumas características relativas à linguagem C, tais como: Portabilidade, instruções acessando diretamente o hardware, Linguagem Estruturada.

A portabilidade faz do programa elaborado, ser executado em qualquer computador, considerando que a referida linguagem esteja instalada no hardware em questão. Além disso esta linguagem é bastante conhecida, sendo encontrada facilmente em sistemas computacionais na área das ciências exatas.

A linguagem C permite, através de instruções específicas uma interação bastante estreita não só com o hardware do computador, bem como dos circuitos ou placas que recebem ou enviam sinais de/para o sistema sob monitoramento. Isto torna o processamento e a conseqüente tomada de decisão bastante rápida, fator primordial para uma aplicação prática de monitoramento.

As vantagens apresentadas, permitiram elaborar o software para supervisão e proteção de uma UPS, utilizando o software Turbo C.

Por tratar-se de uma linguagem seqüencial, isto é, as instruções são processadas uma após a outra, deve-se desde o início impor uma prioridade em termos de qual: compartimento, bloco e análise dos parâmetros será tratado antes.

#### **4.4-) Detalhamento do Programa**

O programa pode ser dividido em dois grandes compartimentos, por ordem de execução:

➤ Definições essenciais ao programa;

- Tela de opções para permitir alteração dos valores máximos de parâmetros analógicos, a serem utilizados pelo programa;
- Seqüência para análise e tomada decisão, a partir do estado dos parâmetros de cada bloco.

A figura 51 representa de forma simplificada esta divisão.

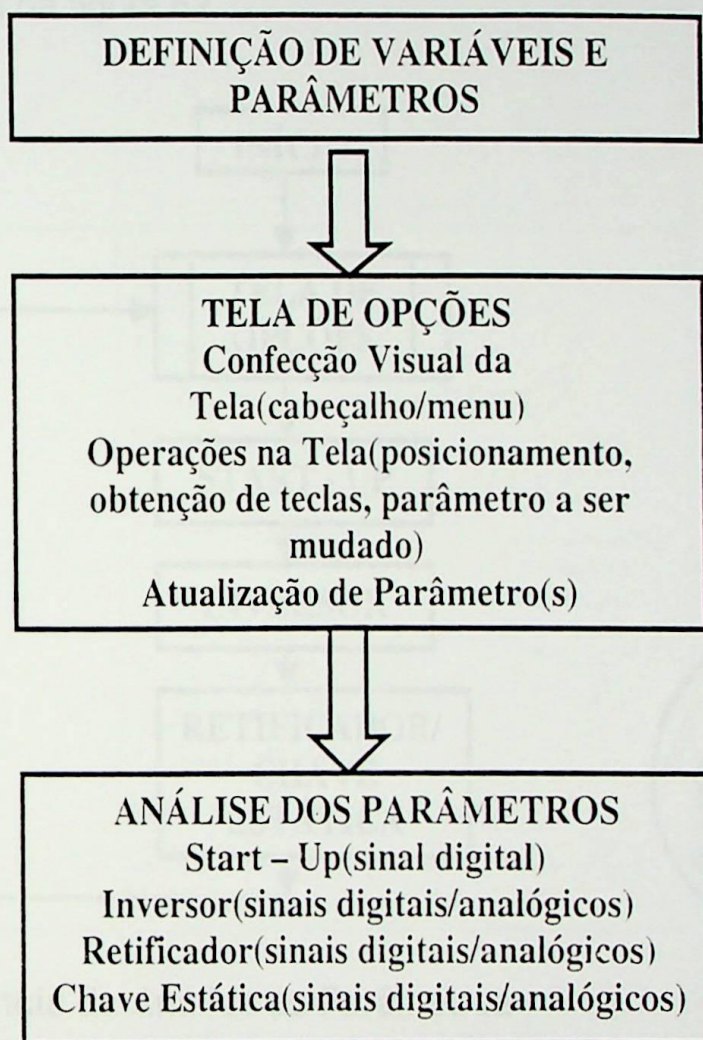


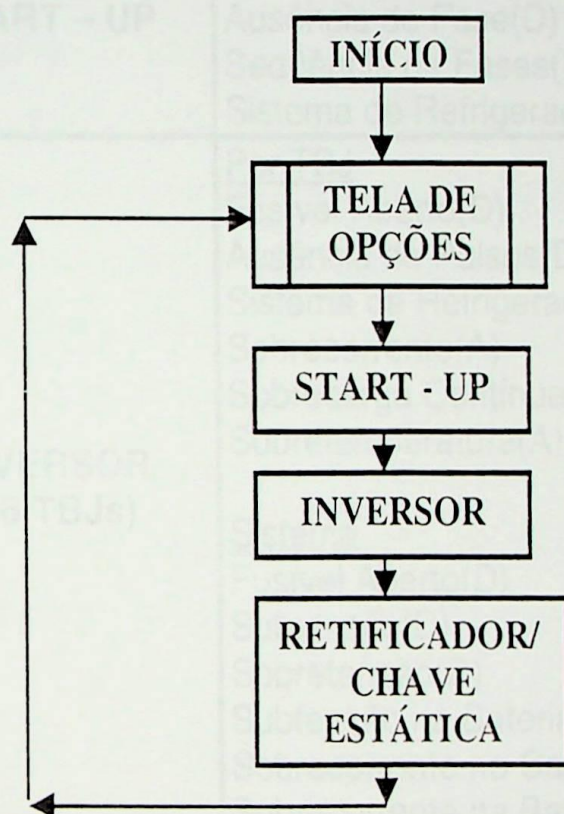
Figura 51 – Compartimentos do Programa

Para melhor abrangência do sistema UPS em termos de análise dos parâmetros digitais e/ou analógicos, dividiu – se o sistema nos seguintes blocos:

- Start–Up;
- Inversor;

- Retificador;
- Chave Estática.

Em termos de prioridade de blocos a serem monitorados, a seqüência anterior foi obedecida de forma decrescente, conforme mostra o fluxograma da figura 52.

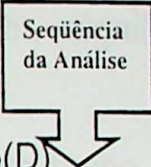
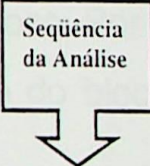
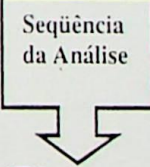


**Figura 52 – Seqüência de Análise de Parâmetros**

A partir desta divisão, escolheu – se um conjunto de parâmetros físicos, que pudessem refletir o funcionamento adequado de cada bloco. Os parâmetros escolhidos foram:

- Tensão;
- Corrente;
- Temperatura.

O tipo de sinal que esses parâmetros podem assumir é: Analógico(A) ou Digital. A tabela 5 mostra de forma resumida, quais as variáveis que refletem o estado do referido bloco.

BLOCO	PARÂMETRO DE FALHA
<p><b>START – UP</b></p>	<p>Alimentação nos Drives(D)                      Subtensão(D)                      Ausência de Fase(D)                      Seqüência de Fases(D)                      Sistema de Refrigeração(D)</p> <div style="text-align: right;">  </div>
<p><b>INVERSOR (06 TBJs)</b></p>	<p><u>Por TBJ</u>                      Fusível Aberto(D)                      Ausência de Pulsos(D)                      Sistema de Refrigeração(D)                      Sobrecorrente(A)                      Sobrecarga Contínua(A)                      Sobretemperatura(A)</p> <p><u>Sistema</u>                      Fusível Aberto(D)                      Subtensão(D)                      Sobretensão(D)                      Subtensão na Bateria(D)                      Sobrecorrente no Capacitor(A)                      Sobrecorrente na Bateria(A)</p> <div style="text-align: right;">  </div>
<p><b>RETIFICADOR (06 SCRs) + CHAVE ESTÁTICA (02 SCRs)</b></p>	<p><u>Por SCR</u>                      Fusível Aberto(D)                      Ausência de Pulsos(D)                      Circuito de Disparo(D)                      Sistema de Refrigeração(D)                      Condução Reversa(A)</p> <p><u>Sistema</u>                      Sobrecorrentes Sucessivas(A)</p> <div style="text-align: right;">  </div>

**Tabela 5 – Parâmetros de Falha Analisados pelo Programa**

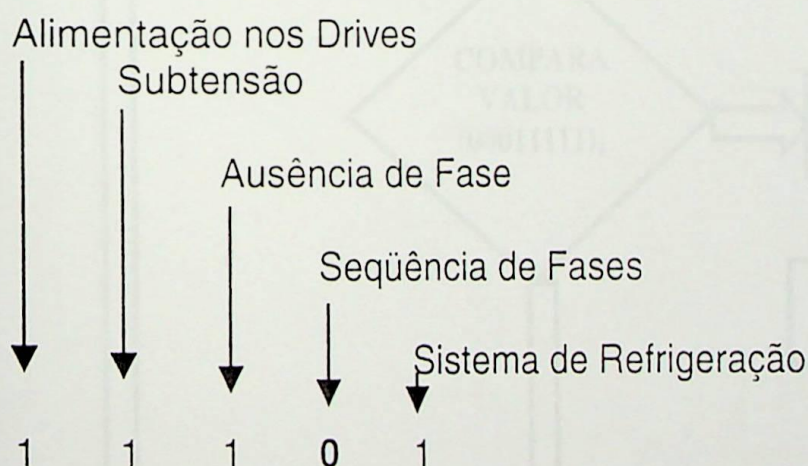
A palavra Sistema empregada na tabela 5, engloba todos os parâmetros que não estão associados diretamente a qualquer dos 12

semicondutores de potência(TBJs e SCRs) checados, e sim a outros componentes(Bateria e Capacitor) bem como parâmetros que são frutos do desempenho global do referido bloco.

O formato do sinal analógico a ser analisado pelo programa é o próprio valor numérico, ou seja, os valores máximos, alterados ou não pelo operador via Tela de Opções, são comparados com os respectivos valores e, caso o valor lido for maior, significa que ocorreu uma falha.

Por exemplo, caso o valor máximo de sobrecorrente fosse definido em 150 °C, se a leitura feita da temperatura do SCR 4 for de 165 °C, o programa irá sinalizar que tal falha ocorreu.

Já o formato do sinal digital é de um byte(08 bits), onde cada bit representa o estado normal(nível lógico 1) ou não(nível lógico 0) de um parâmetro, pertencente a um determinado bloco. Por exemplo, o byte  $(01101)_2$  referente à informação do estado do bloco Start – Up, indica que existe falha na seqüência de fase.



Em qualquer dos formatos de sinal analisado pelo programa, o mesmo irá efetuar os procedimentos abaixo:

- Envio de Sinal Digital ao Circuito de Proteção;
- Aviso da Falha Visualizada na Tela;

➤ Código de Falha Armazenado de Forma Matricial.

Na figura 53 está representado o fluxograma do programa, relativo aos parâmetros de start-up da UPS. Pode-se observar que, enquanto qualquer dos parâmetros não estiver de acordo, o respectivo bit assume nível lógico 0, evitando que o sistema comece a funcionar.

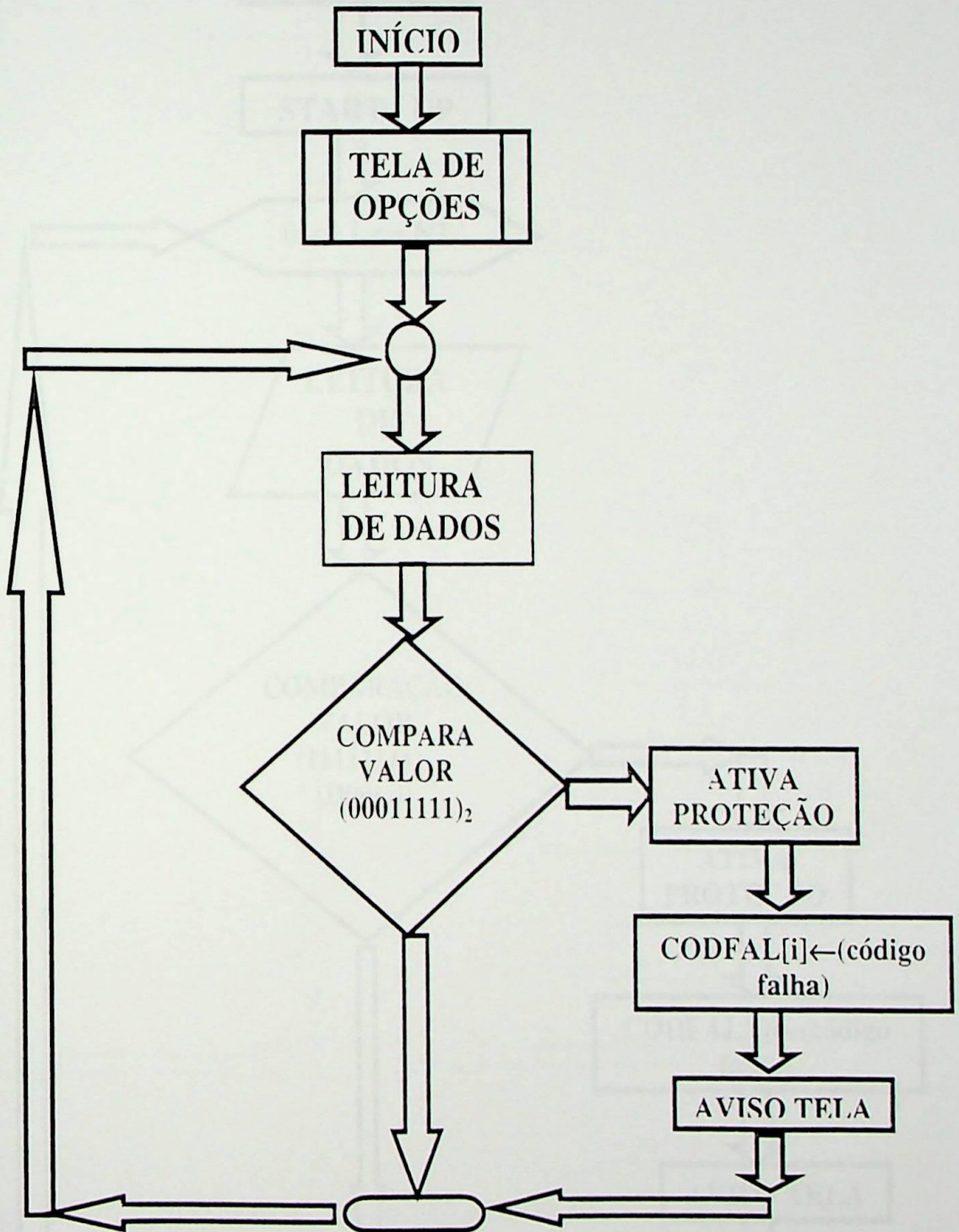


Figura 53 – Verificação dos Parâmetros de Start-Up

Figura 54 – Verificação dos Parâmetros Digitais dos Blocos

Os fluxogramas das figuras 54 e 55 ilustram, de forma resumida, a seqüência de Leitura→Processamento→Tomada de Decisão dos demais blocos, tanto para parâmetros digitais como analógicos.

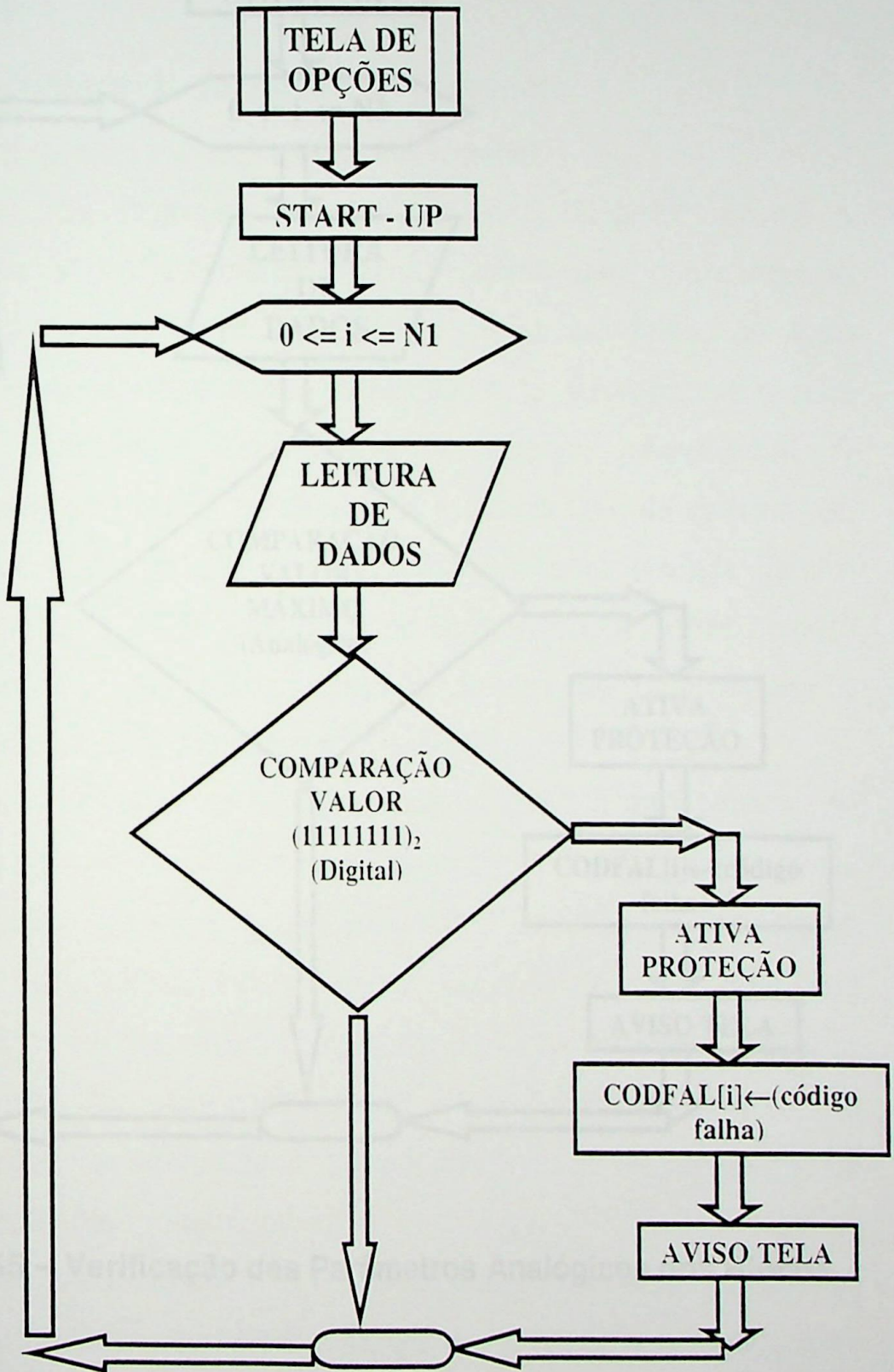


Figura 54 – Verificação dos Parâmetros Digitais dos Blocos

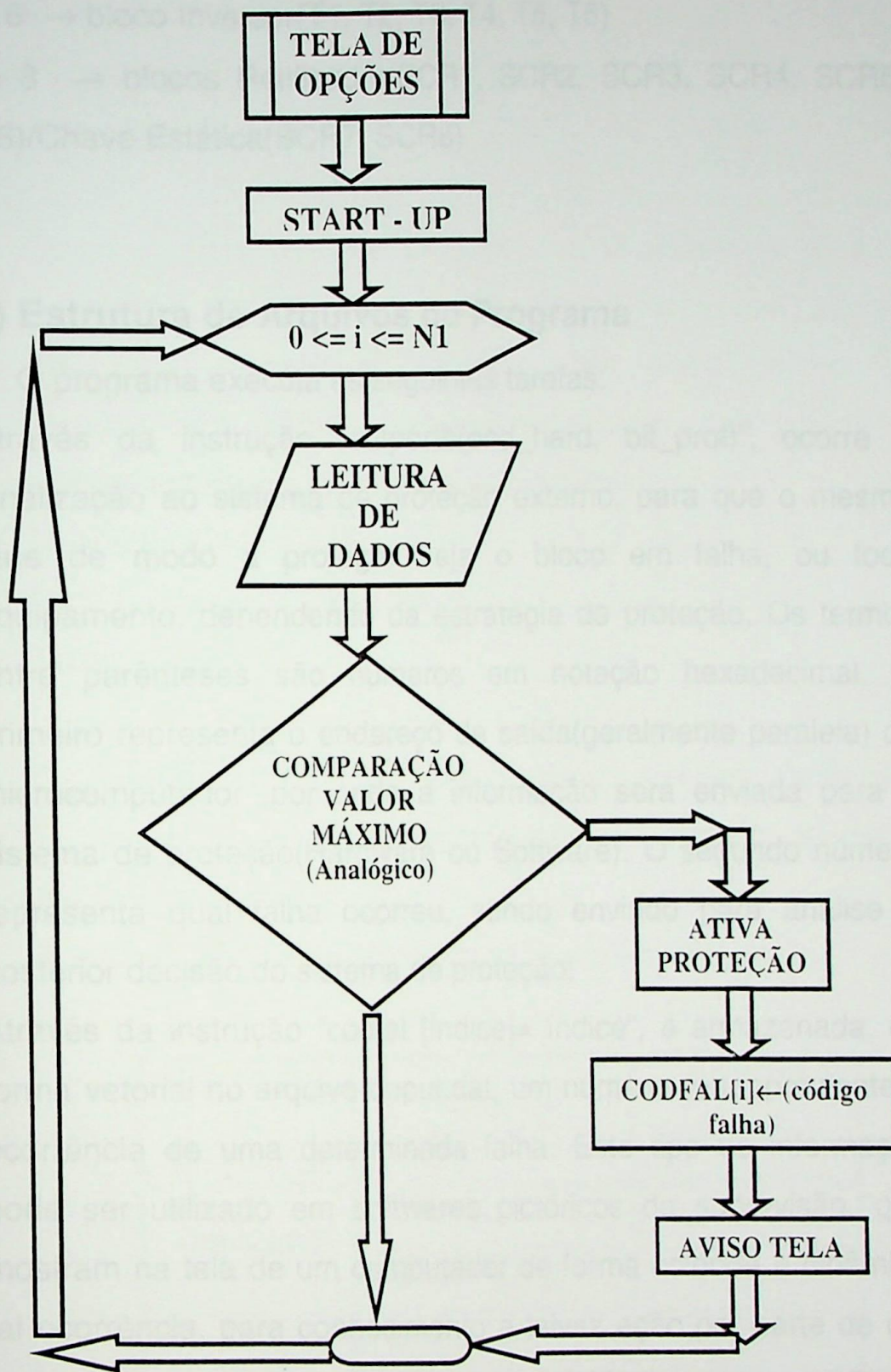


Figura 55 – Verificação dos Parâmetros Analógicos dos Blocos

A estrutura de repetição irá ocorrer tantas vezes, de acordo com o bloco que estiver sendo analisado, ou seja:

N1 = 6 → bloco Inversor(T1, T2, T3, T4, T5, T6)

N1 = 8 → blocos Retificador(SCR1, SCR2, SCR3, SCR4, SCR5, SCR6)/Chave Estática(SCR7, SCR8)

#### 4.5-) Estrutura de Arquivos do Programa

O programa executa as seguintes tarefas:

- Através da instrução “outportb(end\_hard, bit\_prot)”, ocorre a sinalização ao sistema de proteção externo, para que o mesmo atue de modo a proteger seja o bloco em falha, ou todo equipamento, dependendo da estratégia de proteção. Os termos entre parênteses são números em notação hexadecimal. O primeiro representa o endereço de saída(geralmente paralela) do microcomputador ,por onde a informação será enviada para o sistema de proteção(Hardware ou Software). O segundo número representa qual falha ocorreu, sendo enviado para análise e posterior decisão do sistema de proteção;
- Através da instrução “codfal [índice]= índice”, é armazenada, de forma vetorial no arquivo ouput.dat, um número correspondente a ocorrência de uma determinada falha. Este tipo de informação pode ser utilizado em softwares pictóricos de supervisão, que mostram na tela de um computador de forma colorida e dinâmica tal ocorrência, para conhecimento e talvez ação por parte de um operador, remotamente localizado, envolvido numa supervisão de operação de equipamentos.
- Através de instruções “cprintf(“””, pode-se visualizar na tela do microcomputador mais próximo do equipamento, uma mensagem de falha, contendo as seguintes informações:

- ✓ Qual bloco
- ✓ Componente(s)(TBJ ou SCR)
- ✓ Tipo de falha ocorrida

Como se trata de uma simulação de situações de falha do equipamento, é imperativo que o programa receba os referidos sinais. Para informar ao mesmo quais sinais devem ser analisados, são utilizados arquivos específicos contendo dados que representam três estados de funcionamento da UPS:

- Nenhuma Falha;
- Uma Falha(digital e analógica);
- Todas as falhas.

Os arquivos associados a esses estados recebem a extensão .DAT, sendo cada estado representado um arquivo de dados digital e outro analógico.

A tabela 6 mostra os nomes de cada arquivo utilizado.

OCORRÊNCIA DE FALHA	NOME DO ARQUIVO	TIPO DE DADO
Nenhuma	SEMFALD.DAT	Digital
	SEMFALA.DAT	Analógico
Uma	UMAFALD.DAT	Digital
	UMAFALA.DAT	Analógico
Todas	TODAFALD.DAT	Digital
	TODAFALA.DAT	Analógico

Tabela 6 – Arquivos de Dados Usados na Simulação

A figura 56 ilustra, de forma simplificada, a interação entre os arquivos de dados e o programa de supervisão.

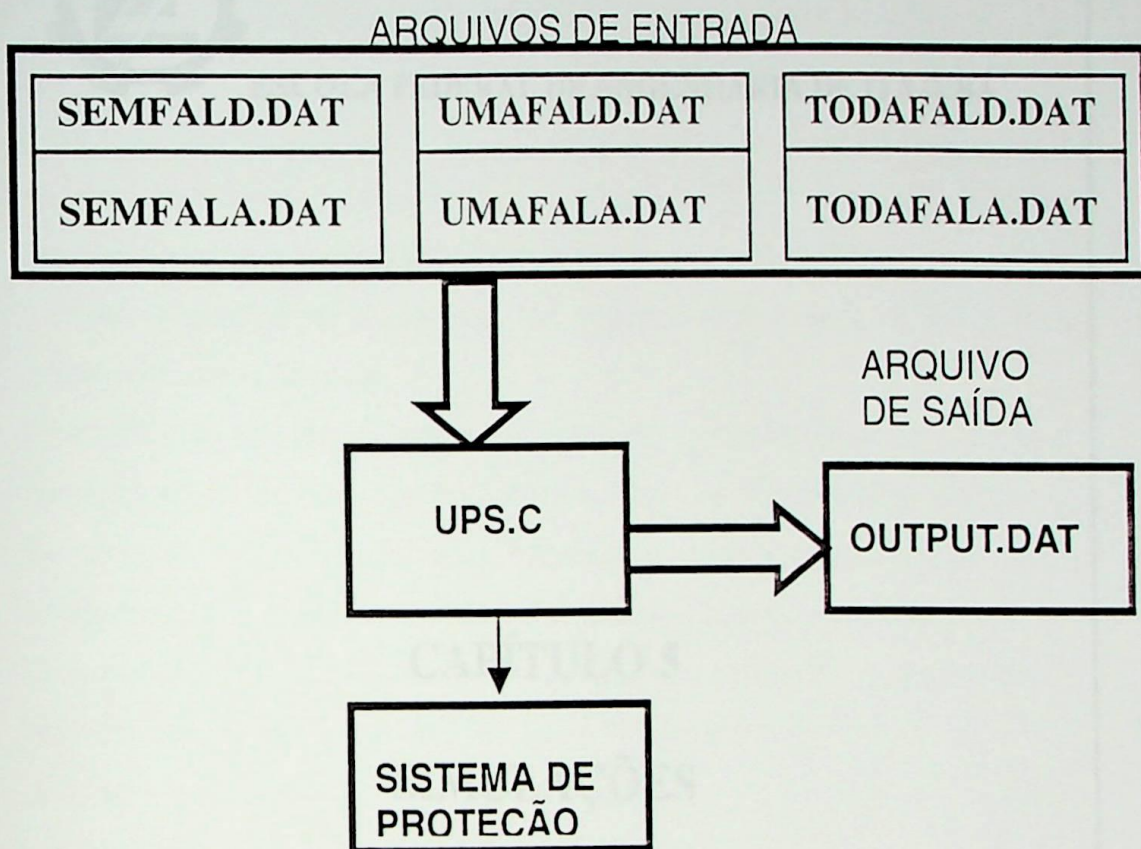


Figura 56 – Interação Entre os Arquivos

A listagem do programa UPS.C(programa de verificação da UPS), bem como dos arquivos de dados(arquivos.DAT) está mostrada no Apêndice A.

O capítulo seguinte vai descrever as simulações realizadas para validação do programa.



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## CAPÍTULO 5 SIMULAÇÕES

## CAPÍTULO 5 – SIMULAÇÕES

### 5.1-) Considerações Iniciais

O presente capítulo objetiva descrever e apresentar os resultados, das simulações do programa proposto.

Foram feitas simulações de hardware e software. A de hardware foram feitas nos circuitos apresentados no Capítulo 3. O presente capítulo apresenta a estratégia de validação do programa de supervisão:

- Tornar o programa elaborado, um arquivo executável(UPS.EXE) sem erros e/ou advertências;
- Analisar, em termos de tempo de execução o desempenho do programa, bem como os blocos que compõem a estrutura dos circuitos eletrônicos de uma U.P.S. típica, os quais recebem e analisam os sinais digitais e analógicos dos parâmetros, refletindo a integridade do sistema;
- Comparar todos os tempos acima, executando o mesmo programa em 03 microcomputadores tipo IBM-PC, cujas CPUs foram:
  - ✓ 80486 – 100 MHz
  - ✓ PENTIUM – 133 MHz
  - ✓ PENTIUM – 233 MHz.

### 5.2-) Estratégia da Simulação

Na presente simulação, o ambiente computacional utilizado foi:

- Sistema Operacional D.O. S., Versão 7.0 - Microsoft Corporation
- Software TURBO – C, Versão 3.0 – Borland International Incorporated

A simulação constou da leitura de arquivos de dados (**nome do arquivo**).DAT, por parte do programa principal(UPS.C). Cada leitura(digital ou analógica) correspondente a um parâmetro de supervisão é comparado com um valor limite e, em caso de não - igualdade, para os dados digitais, ou se forem superiores aos valores analógicos máximos, alteráveis via menu inicial, o programa toma as seguintes atitudes:

- envia um sinal de aviso de falha ao hardware de proteção;

- mostra na tela do computador, informações relativas à(s) falha(s) ocorrida(s).

Como foi citado anteriormente, os dados analógicos são comparados com valores máximos, que podem ser alterados pelo usuário, através do menu de opções, constando de sete parâmetros, que podem ser selecionados e/ou alterados, via teclas específicas do teclado, conforme abaixo:

- <TAB> permite seleção/confirmação do parâmetro
- <ESC> cancela o valor digitado, mantendo o valor anterior
- <ENTER> confirma valores mostrados no menu, além de iniciar a supervisão dos dados digitais e analógicos, contidos nos arquivos de dados, por parte do programa

O quadro 1, reproduz o menu anteriormente citado.

### SUPERVISÃO E PROTEÇÃO DE UM UPS

Alterar valor(es) máximo(s),selecione-o(os) via tecla <TAB> e, digite- o(s).

Pressione tecla <ESC> para voltar ao valor máximo anterior.

Pressione tecla <ENTER> para confirmar todos os valores(programa em execução).

PARA FINALIZAR O PROGRAMA, PRESSIONE NOVAMENTE TECLA <ESC>.

#### VALORES MÁXIMOS

ATRASSO START-UP:	1.000 [ms]
SOBRECORRENTE(TBJ):	100.000 [A]
SOBRECARGA CONTINUA(TBJ):	100.000 [A]
SOBRETEMPERATURA(TBJ):	120.000 [°C]
SOBRECARGA(CAPACITOR)	100.000 [A]
SOBRECARGA(BATERIA)	100.000 [A]
CONDIÇÃO REVERSA(SCR):	100.000 [A]
Imáx NA CARGA:	100.000 [A]

Quadro 1 – Tela de Seleção de Valores

Na ocorrência de uma ou mais falhas, o programa avisa ao operador, de forma visual:

- qual bloco ocorreu a falha
- qual componente
- tipo de falha.

O quadro 2 reproduz um exemplo de aviso na tela ao operador na ocorrência de mais de uma falha:

**FALHA NO BLOCO INVERSOR**  
**falha no TBJ 1**  
**Falha de Sobrecorrente(SINAL ANALOGICO)**

**FALHA NO BLOCO INVERSOR**  
**Falha de Fusível Aberto(SINAL DIGITAL)**  
**FALHA NO BLOCO CAPACITOR**  
**Falha de Sobrecorrente(SINAL ANALOGICO)**

**FALHA NO BLOCO RETIFICADOR**  
**falha no SCR 1**  
**Falha de Fusível Aberto(SINAL DIGITAL)**  
**FALHA NO BLOCO RETIFICADOR**  
**falha no SCR 1**  
**Falha de Condução Reversa(SINAL ANALOGICO)**

**FALHA NO BLOCO RETIFICADOR**  
**Falha de Sobrecorrentes Sucessivas(SINAL ANALOGICO)**

#### Quadro 2 – Exemplo de Aviso ao Operador

No apêndice D são mostradas outros exemplos de telas, sinalizando ocorrência de outras falhas do UPS.

### 5.3-) Tempos de Execução do Programa

Além da elaboração do programa, foi feito um estudo de tempo, relativo à duração que o hardware levou para executar a verificação dos parâmetros dos blocos componentes da UPS.

Inicialmente, o programa exibe a tela inicial que permite ao usuário digitar os valores máximos analógicos que o programa irá se basear para análise dos referidos parâmetros. A partir da confirmação destes valores, é contado o tempo de execução dos blocos como um todo, além de cada bloco em separado, na ocorrência de:

- Nenhuma Falha;

- Uma Falha;
- Todas as Falhas.

Para o estudo de tempos foi utilizado o programa [14], onde duas rotinas `crono_start()` e `crono_stop()` são colocadas no início e término do fragmento do programa a ser temporizado. A diferença entre os dois tempos, será o tempo que o referido fragmento do programa foi executado.

A tabela 7 a seguir, mostram os tempos médios de execução tanto do programa, como relativos a cada bloco do UPS, em termos de:

- Número de Falhas Consideradas
- Hardware Utilizado

Vale ressaltar que, cada valor da tabela 7 é resultante da média aritmética de 10 leituras efetuadas, sendo que todos os valores estão apresentados no Apêndice B.

Ocorrência de Falhas	HARDWARE		
	80846	Pentium	Pentium II
NENHUMA	0,0258169	0,0283845	0,03330535
UMA	0,030036	0,0283845	0,0483379
TODAS	0,532604	0,3959857	0,4183978

**Tabela 7 - Tempos Médios de Execução do Programa**

Da tabela 7 pode –se observar que:

- A ordem de grandeza dos valores é a mesma;
- Não houve ganho substancial de tempo com atualização do hardware.

Os gráficos das figuras 57 e 58 a seguir, mostram a tendência dos tempos coletados na prática, que levaram aos valores médios da tabela 7.

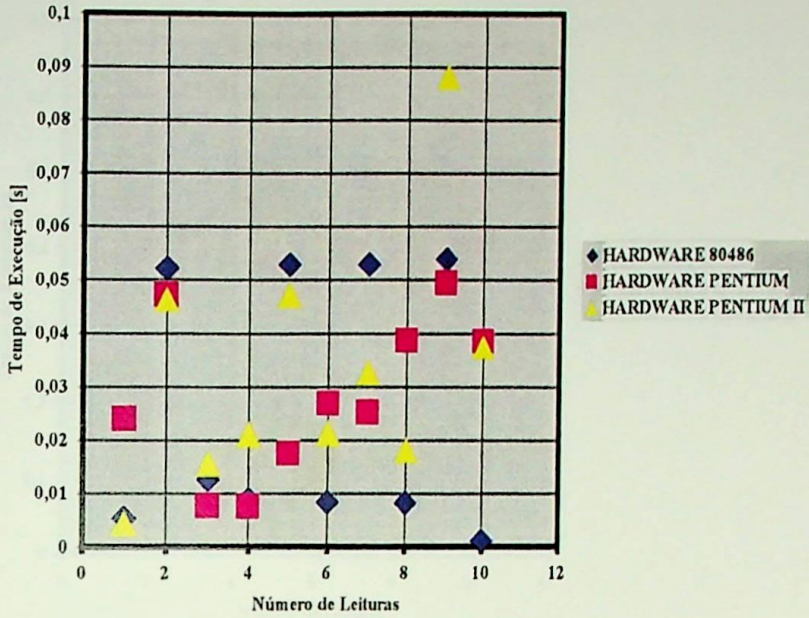


Figura 57 – Sem Falha

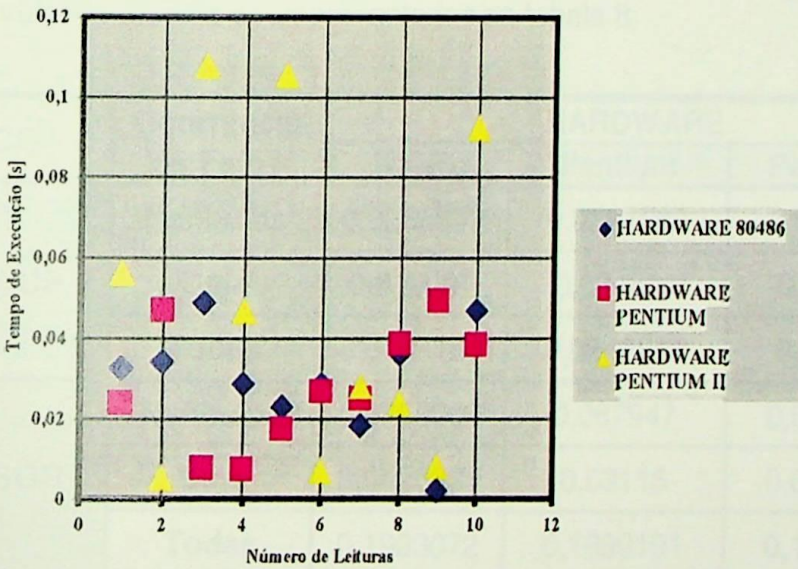


Figura 58 – Uma Falha

Comparando os gráficos anteriores, vê-se que:

- Na figura 57, os tempos variam de forma similar, variação de 0,01 a 0,05 s;
- Na figura 58, apenas os tempos do hardware Pentium II não variam de forma similar em relação aos demais, entre estes variação de ,004 a ,05 s.

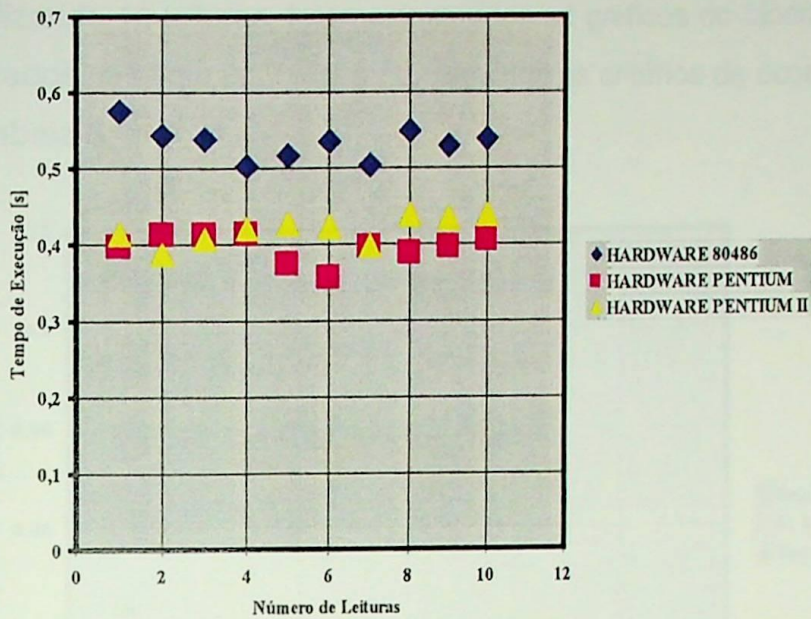


Figura 59 – Todas Falhas

Para o gráfico da figura 59, pode-se observar que ocorre uma tendência constante dos tempos.

Também foi feito uma coleta de tempos para todos os blocos da UPS, cujos valores médios são apresentadas na tabela 8.

BLOCO	Ocorrência de Falha	HARDWARE		
		80486	Pentium	Pentium II
START-UP	Nenhuma	0,0285959	0,0291568	0,0270542
	Uma	0,019094	0,034631	0,020511
	Todas	0,0088163	0,0346663	0,021712
INVERSOR	Nenhuma	0,0298006	0,067947	0,0175924
	Uma	0,0428821	0,03115	0,0375684
	Todas	0,1903072	0,1999191	0,1878184
SISTEMA	Nenhuma	0,019738	0,022892	0,0334049
	Uma	0,02596	0,01772	0,0219265
	Todas	0,228712	0,0863563	0,0668751
RETIFICADOR/ CHAVE ESTÁTICA	Nenhuma	0,047947	0,025123	0,0330617
	Uma	0,02808	0,05439	0,0404423
	Todas	0,2477214	0,2206665	0,2357292

Tabela 8 –Tempos Médios de Execução dos Blocos

Utilizando 10 leituras, foram construídos os gráficos do bloco START-UP, mostrados nas figuras 60, 61 e 62, seguindo os critérios de ocorrência de falha da tabela 8.

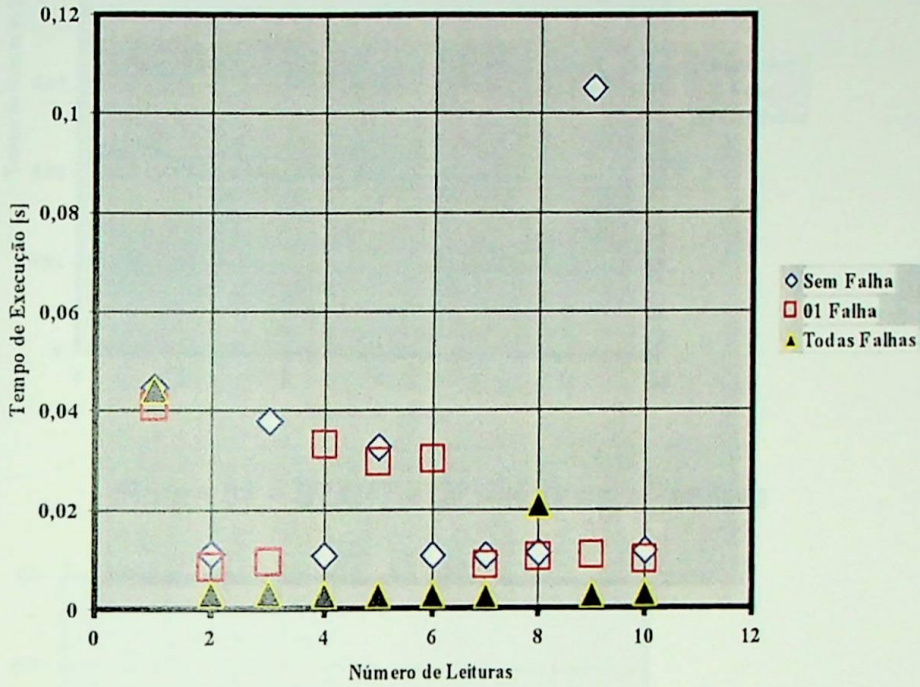


Figura 60 – START-UP – Hardware 80486

Analisando a figura 60, pode-se observar que:

- Apenas os tempos relativos a “Todas Falhas” não acompanharam os demais tempos;
- Variação dos tempos, intervalo de 0,0 s a 0,042 s.

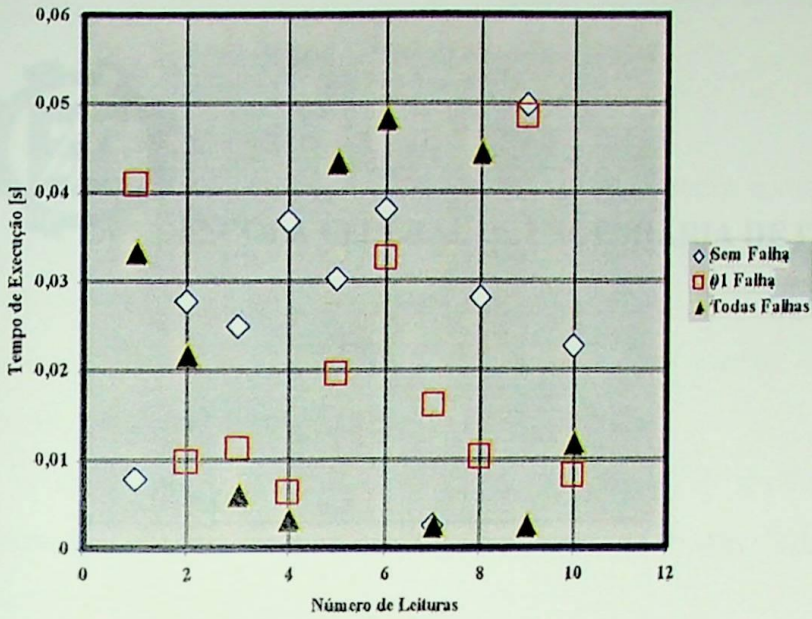


Figura 61 – START – UP – Hardware Pentium

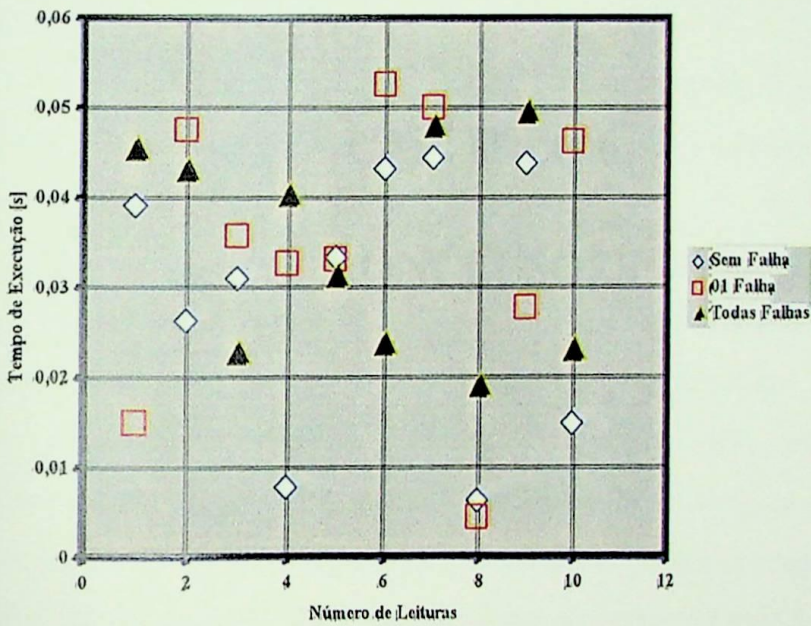


Figura 62 – START – UP – Hardware Pentium II

A partir das figuras 61 e 62, vê-se que:

- Comportamento constante dos dados a partir da 6ª leitura;
- Variação dos tempos, intervalo 0,01 s a ,04 s.



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## CAPÍTULO 6

### CONCLUSÕES

## CAPÍTULO 6 – CONCLUSÕES

O objetivo do trabalho foi, a partir de um sistema contendo circuitos de potência, como é o caso do UPS, elaborar um programa de computador para, a partir da análise de sinais analógicos e digitais, acionar o sistema de proteção no sentido de preservar o perfeito funcionamento do referido sistema, evitando problemas técnico-financeiros.

É permitido ao usuário do programa, modificar os valores analógicos máximos, que serão levados em conta na verificação dos dados coletados pelo hardware.

Na ocorrência de uma ou mais falhas, tal fato pode ser visto e detalhado na tela do computador, via mensagens discriminando: qual bloco, tipo de falha e em qual componente ocorreu.

Pela quantidade de variáveis a serem monitoradas, optou-se por elaborar arquivos de dados, de forma a permitir rápida e confiável validação da estratégia de software proposta.

Uma outra contribuição do trabalho, foi a de fazer um estudo do tempo de execução, não só do programa em si, como também dos blocos que compõem o UPS, em termos de verificação das variáveis sob supervisão.

Observou-se que, os tempos ficaram na faixa de décimos ou até centésimos de segundos podendo ser utilizados na prática, pois os tempos de execução de programas utilizados por sistemas de supervisão industriais (CLPs, SDCDs, Multiloop, etc) possuem esta mesma ordem de grandeza.

O fator agravante no caso é os tempos de atraso inerentes aos componentes eletrônicos dos circuitos de leitura de dados.

A principal contribuição do presente trabalho, é a elaboração de um programa específico para detecção de falhas num circuito de potência, além de executar a verificação do tempo decorrido do referido software.

Como sugestão para futuros trabalhos de pesquisa ou mesmo de dissertação de mestrado, dando seqüência ao trabalho desenvolvido, pode-se enumerar:

- Aperfeiçoamento e otimização do software proposto, a fim de se obter menor tempo de processamento;
- Integração deste software com um sistema de supervisão/proteção (telas amigáveis em Visual C);
- Implementação de hardware específico para realizar leitura dos parâmetros a serem monitorados pelo programa;
- Implementação de hardware específico para realizar a proteção de uma UPS, a partir da execução do software elaborado;
- Implementação em Tempo Real utilizando DSP (“Digital Signal Processor”).

## APÊNDICE A



ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## APÊNDICE A

## APÊNDICE A – PROGRAMA FONTE EM LINGUAGEM C

```

// programa da supervisao e protecao da UPS
-
-
//***** ARQUIVOS CABEÇALHO *****
-
#include <bios.h>
#include <conio.h>
#include <crono.h>
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
-
// ***** CONSTANTES USADAS NO MENU *****
-
#define BACKSPACE  0x08 // valor hexa teclas
#define DECIMAL0   0x30
#define DECIMAL9   0x39
#define ENTER      0x0d
#define ESC        0x1b
#define PONTO      0x2e
#define TAB        0x09
#define t " "
-
-
// ***** VALORES M:IXIMOS INICIAIS *****
-
#define DELTAT      1. // tempo start-up
#define SOBRECORRMAX 100. // sobrecorrente maxima TBJ
#define SOBRECORRMAX1 100. // sobrecorrente maxima capacitor
#define SOBRECORRMAX2 100. // sobrecorrente maxima bateria
#define SOBRECARGACON 100. // sobrecarga cont:nuua maxima TBJ
#define TMAX        120. // temperatura m:axima TBJ
#define CONDREVMAX  100. // condi:cao reversa m:axima SCRs
#define ILMAX       100. // valor m:aximo corrente carga
-
// ***** CONSTANTES USADAS NO PROGRAMA *****
-
#define VERDADE     1
#define FALSO       0
#define NJANELAS    8 // n:umero valores m:aximos mostrados
-
// ***** DEFINI:CAO CONSTANTES HARDWARE *****
-
#define MASCARA1    0x1f // par:ametros start-up - (00011111)2 (31)10
#define MASCARA2    0x03 // par:ametros inversor - (0011)2 (03)10
#define MASCARA3    0x3f // par:ametros sistema - (00111111)2 (63)10

```

```

#define MASCARA4    0x0f // parâmetros retificador - (00001111)2 (255)10
#define NMAXCOD     3000 // número total parâmetros supervisionados
UPS
#define PORTOUT     0x378 // endereço porta paralela(LPT1) saída
hardware
:
:
: // ***** DEFINIÇÃO CONSTANTES FISICAS PROGRAMA *****
:
:const float TLIMITE1 = 2.5e0-3 ; /* tempo ocorrência sobrecorrente */
:const float TLIMITE2 = 5.e0-3 ; /* tempo ocorrência sobrecorrentes consecut
:const float VTO     = 1.2   ; /* tensão linearizada SCR */
:const float RT      = 5.0E-3 ; /* resistência linearizante SCR */
:const float ROJC    = 0.02  ; /* resistência junção-corpo SCR */
:
:
: int num jan = FALSO ; // INICIO NA 1a JANELA
: double tempo ;
: float valor[NJANELAS]= // VALOR DIGITADO
:     { DELTAT, SOBRECORRMAX, SOBRECARGACON, TMAX,
:     SOBRECORRMAX1,
:     SOBRECORRMAX2, CONDREVMAX, ILMAX };
: unsigned xi[NJANELAS]= { VERDADE+6 , VERDADE+6 , VERDADE+6 ,
: VERDADE+6 ,
:     , yi[NJANELAS]= { VERDADE+7 , VERDADE+9 , VERDADE+11 ,
: VERDADE+13 , VERD
:     , xf[NJANELAS]= { VERDADE+79 , VERDADE+79 , VERDADE+79 ,
: VERDADE+79 , VERD
:     , yf[NJANELAS]= { VERDADE+7 , VERDADE+9 , VERDADE+11 ,
: VERDADE+13 , VERD
:     ; //coordenadas
: char *menutxt[]={ "      ATRASO START-UP:      %7.3f [ms]", // menu v
:     "      SOBRECORRENTE(TBJ):      %7.3f [A] ",
:     "      SOBRECARGA CONTINUA(TBJ): %7.3f [A] ",
:     "      SOBRETENPERATURA(TBJ):   %7.3f [oC]",
:     "      SOBRECARGA(CAPACITOR)    %7.3f [A] ",
:     "      SOBRECARGA(BATERIA)      %7.3f [A] ",
:     "      CONDIÇÃO REVERSA(SCR):    %7.3f [A] ",
:     "      Imáx NA CARGA:           %7.3f [A] "
:     };
:
: // ***** EXIBE DADO ANTERIOR/DADO SER MUDADO ** */
:
: void pos jan(int num jan) // exhibe dado anterior e dado a ser mudado
: {
: {
:
: int anterior = (num jan > FALSO) ? (num jan - 1) : (VERDADE+6) ;
:
: window ( xi[anterior] , yi[anterior] , xf[anterior] , yf[anterior] ) ; // dado
: textbackground(BLACK) ; textcolor (GREEN) ;
: cprintf( menutxt[anterior] , valor[anterior] ) ;

```

```

:
:window ( xi[num jan] , yi[num jan] , xf[num jan] , yf[num jan] ) ;// dado a se
:|textbackground(BLACK); textcolor (YELLOW+BLINK);
:|cprintf( menutxt[num jan] , valor[num jan] ) ;
:|}
:
:***** OBTÉM TECLA *****
:
:|void getnum( char *buff )
:|{
:|char tecla, *ptr= buff;
:|struct text info ti;
:|int dígitos, width, ponto = FALSO;
:
:|gettextinfo( &ti );
:|width= ti.winright - ti.winleft; // dimensão campo dados
:
:|*ptr= '\0'; // insere NULL
:|dígitos = FALSO;
:
:|do {
:|tecla = getch();
:
:|if( (tecla== PONTO) && (ponto == VERDADE) ) continue;
:
:|if(( tecla==PONTO) ((DECIMAL0 <= tecla) && (tecla <= DECIMAL9) ) &&
:(di
:|// tecla é número ou ponto decimal ou final campo
:|{
:|*ptr++= tecla; *ptr= '\0';
:|if( tecla == PONTO ) ponto = VERDADE;
:|putch( tecla ); // mostra dado
:|dígitos++;
:|}
:|else if( (tecla == BACKSPACE) && (dígitos > FALSO) ) // Correção dados
:|{
:|if( *(ptr-1) == PONTO ) ponto= FALSO; // final ponteiro
:
:|*--ptr= '\0';
:|gotoxy( wherex()-1, wherey() ); // limpa campo dado
:|putch( ' ' );
:|gotoxy( wherex()-1, wherey() );
:|dígitos--;
:|}
:
:|} while( (tecla!= TAB) && (tecla!=ESC) && (tecla!= ENTER) ); // confirmar o
:
:|if( tecla == ESC ) buff[0]= '\0';
:|}

```

```

|
|/|***** Verificação de Bits Ligados *****
|
|#define BITset( Val, bt ) ( ( Val)&(bt) )==(bt)
|
|/* ***** PROGRAMA PRINCIPAL ***** */
|
|void main (void)
|{
|register unsigned i , índice , qualcomp ;
|unsigned DOdata , tecla , startup , status , *codfal;
|unsigned Executar = VERDADE; // Enquanto VERDADE repetir o LOOP
|PRINCIPAL
|float deltat , statusan , sobrecorrmax , sobreccorrcap , sobreccorrbat ,
| sobreccargacon, condrevmax , tmax , ilmax , itav ,itrms , tc ;
|char str[100]= "" , binário[600]="";
|
|struct text info ti;
|FILE *digital , *analog , *output ; /* arquivos de dados */
|
|if ((digital = fopen("SEMFALD.DAT", "rt")) == NULL)
|//if ((digital = fopen("UMAFALD.DAT", "rt")) == NULL)
|//if ((digital = fopen("TODAFALD.DAT", "rt")) == NULL)
| {
| clrscr() ;
| fprintf(stderr, "\nNao é possível abrir arquivo DIGINPU1.DAT.\n");
| exit(VERDADE);
| }
|if ((analog = fopen("SEMFALA.DAT", "rt")) == NULL)
|//if ((analog = fopen("UMAFALAN.DAT", "rt")) == NULL)
|//if ((analog = fopen("TODAFALA.DAT", "rt")) == NULL)
| {
| clrscr() ;
| fprintf(stderr, "\nNao é possível abrir arquivo ANALOG1.DAT.\n");
| exit(VERDADE);
| }
|
|if ((output = fopen("OUTPUT.DAT", "a+t")) == NULL)
| {
| clrscr() ;
| fprintf(stderr, "\nNao é possível abrir arquivo OUTPUT.DAT.\n");
| exit(VERDADE);
| }
|
|/|***** CABEÇALHO *****
|
|gettextinfo( &ti);
|
|window(VERDADE,VERDADE,VERDADE+79,VERDADE+24) ;

```

```

|clrscr() ; // limpa tela
|textcolor(WHITE) ; textbackground(BLACK) ;
|cprintf("          SUPERVISAO E PROTEÇÃO DE UMA UPS") ;
|
|gotoxy( VERDADE,VERDADE+1 ) ;
|textcolor(YELLOW) ; textbackground(BLACK) ;
|cprintf("Alterar valor(es) máximo(s),selecione-o(os) via tecla <TAB> e, digite
|cprintf("Pressione tecla <ESC> para voltar ao valor máximo anterior.\n\n") ;
|cprintf("Pressione tecla <ENTER> para confirmar todos os valores(programa
em e
|textcolor(CYAN) ; textbackground(BLACK) ;
|cprintf("PARA FINALIZAR O PROGRAMA, PRESSIONE NOVAMENTE
TECLA <ESC>.\n\n\n");
|
|//***** MENU(tela c/ valores iniciais) *****
|
|gotoxy( VERDADE+14,VERDADE+5 ) ;
|textcolor(GREEN) ; textbackground(BLACK) ;
|cprintf("\n\n          VALORES MAXIMOS\n");
|
|do{
|
|// Imprime o Menu
|for( indice= FALSO ; indice <= (VERDADE+6) ; indice++ ) {
| window ( xi[indice] , yi[indice] , xf[indice] , yf[indice] ) ;
| textbackground(BLACK) ; textcolor (GREEN) ;
| clreol();
| cprintf( menutxt[indice] , valor[indice] ) ;
| }
|
|// ***** POSICIONA NUMA JANELA *****
|
|pos jan(num jan) ;
|
|fflush(stdin) ; // limpa buffer entrada dados
| do{
| tecla = getch() ;
| if (tecla == TAB) {
| num jan+= (num jan < NJANELAS-1) ? (VERDADE) : (-NJANELAS+1) ;
| fflush(stdin) ;
| pos jan( num jan ) ;
| }
| } while (tecla == TAB) ;
|
| if( (tecla == PONTO) || ( (DECIMAL0 <= tecla) && (tecla <= DECIMAL9) ) ){
| pos jan(num jan);
| fflush(stdin);
| ungetch(tecla); // Devolve para getnum();
| window( xi[num jan]+33+4 , yi[num jan] , xi[num jan]+42+3 , yf[num jan] );
| getnum(str);

```



```
}
if(!BITset(status,0x002))
{
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
cprintf("\n\r" t "" t "Falha de Subtensao(SINAL DIGITAL)\n\r") ;
/* código falha subtensão */
}
if(!BITset(status,0x004))
{
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+1 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
cprintf("\n\r" t "" t "Falha de Ausência de Fases(SINAL DIGITAL)\n\r") ;-
/* código falha erro ausência fases */
}
if(!BITset(status,0x008))
{
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+2 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
cprintf("\n\r" t "" t "Falha na Seqüência de Fases(SINAL DIGITAL)\n\r")
/* código falha seqüência fases */
}
if(!BITset(status,0x010))
{
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+3 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
cprintf("\n\r" t "" t "Falha no Sistema de Refrigeração(SINAL DIGITAL)\n\r")
/* código falha sistema refrigeração */
}
goto startup;
}
}

delay( deltat ) ; // proporciona atraso (start-up) deltaT milissegundos
```

```

for ( qualcomp = FALSO ; qualcomp <= (VERDADE+4) ; qualcomp++)
{
if( fscanf( digital , "%s" , binario )!= FALSO )
{
for( i= FALSO, status= FALSO; binario[i]!='\0'; i++ )
{
status<<= VERDADE ;
status|= (binario[i]=='1') ? VERDADE : FALSO ;
}
}
else fprintf(stderr, "\n\taErro de Leitura em fscanf: Arquivo %s Linha %d", -
if ( (qualcomp == FALSO) || (qualcomp == (VERDADE + 1)) )
{
status >>= VERDADE+2 ;
}
if ( !BITset(status,MASCARA2) ) // Há erro em algum dos 3 bits?
{
DOdata = VERDADE ; // sinaliza falha bloco INVERSOR
outportb(PORTOUT,DOdata) ;
textcolor (LIGHTRED) ; textbackground(BLACK) ;
cprintf("\n\n\r") ;
cprintf("          FALHA NO BLOCO INVERSOR") ;
switch(qualcomp+VERDADE)
{
case VERDADE      : if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+4 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output
textcolor (LIGHTGREEN) ; textbackground(BLACK) ; -
cprintf("\n\r") ;
cprintf("          falha no TBJ 1") ;
break ;
/* código falha TBJ 1 */
case (VERDADE + 1) : if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+5 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r ") ;
cprintf("          falha no TBJ 2") ;
break ;
/* código falha TBJ 2 */
case (VERDADE + 2) : if( (codfal++) == NULL )
{
exit(1) ;
}
}
}

```

```
        *codfal = VERDADE+6 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf("\n\r") ;
        cprintf("          falha no TBJ 3") ;
        break ;
/* código falha TBJ 3 */
case (VERDADE + 3) : if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+7 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf( "\n\r") ;
    cprintf("          falha no TBJ 4") ;
/* código falha TBJ 4 */
case (VERDADE + 4) : if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+8 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf( "\n\r") ;
    cprintf("          falha no TBJ 5") ;
/* código falha TBJ 5 */
case (VERDADE + 3) : if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+9 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf( "\n\r") ;
    cprintf("          falha no TBJ 6") ;
/* código falha TBJ 6 */
}
if ( !BITset(status,0x001) )
{
    if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+10 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf("\n\r" t "" t "Falha de Fusível Aberto(SINAL DIGITAL)\n\r") ;
    /* código falha fusível aberto */
}
```

```

if ( !BITset(status,0x002) )
{
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+11 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r" t "" t "Falha de Ausência de Pulsos(SINAL DIGITAL)\n\r"
/* código falha ausência pulsos */
}
if ( !BITset(status,0x004) )
{
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+12 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r" t "" t "Falha no Sistema de Refrigeração(SINAL DIGITAL)
/* código falha sistema refrigeração */
}

// VERIFICA STATUS ANALOGICO CADA TBJ - PONTE INVERSORA *****

/* leitura sinais analógicos(sobrecorr., sobrecarga contínua, lav, Irms, tc)
fscanf( analog , "%f" , &statusan ) ;
if ( statusan > sobrecorrmax )
{

{
DOdata = VERDADE ; // falha bloco INVERSOR
outputb(PORTOUT,DOdata) ; // sinaliza atuação proteção
textcolor (LIGHTRED) ; textbackground(BLACK) ;
cprintf("\n\r" ) ;
cprintf("          FALHA NO BLOCO INVERSOR") ;
switch(qualcomp+VERDADE)
{
case VERDADE      : if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+13 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r" ) ;
cprintf("          falha no TBJ 1") ;
break ;
}
}
}

```

```
/* código falha TBJ 1 */
case (VERDADE + 1) : if( (codfal++) == NULL )
    {
        exit(1);
    }
    *codfal = VERDADE+13 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf("\n\r") ;
    cprintf("          falha no TBJ 2") ;
    break ;
/* código falha TBJ 2 */
case (VERDADE + 2) : if( (codfal++) == NULL )
    {
        exit(1);
    }
    *codfal = VERDADE+14 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf("\n\r") ;
    cprintf("          falha no TBJ 3") ;
    break ;
/* código falha TBJ 3 */
case (VERDADE + 3) : if( (codfal++) == NULL )
    {
        exit(1);
    }
    *codfal = VERDADE+15 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf("\n\r") ;
    cprintf("          falha no TBJ 4") ;
/* código falha TBJ 4 */
case (VERDADE + 4) : if( (codfal++) == NULL )
    {
        exit(1);
    }
    *codfal = VERDADE+16 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf( "\n\r") ;
    cprintf("          falha no TBJ 5") ;
/* código falha TBJ 5 */
case (VERDADE + 3) : if( (codfal++) == NULL )
    {
        exit(1);
    }
    *codfal = VERDADE+17 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
```

```
        cprintf( "\n\r" );
        cprintf("          falha no TBJ 6" );
/* código falha TBJ 6 */
}
if( (codfal++) == NULL )
{
    exit(1);
}
*codfal = VERDADE+18;
fwrite ( codfal , sizeof(codfal) , VERDADE , output );
textcolor (LIGHTGREEN); textbackground(BLACK);
cprintf("\n\r" t "" t "Falha de Sobrecorrente(SINAL ANALOGICO)\n\r" );
/* código falha sobrecorrente */

fscanf( analog , "%f" , &statusan );
if ( statusan > sobrecargacon )
{
    DOfdata = VERDADE ; // falha bloco INVERSOR
    outportb(PORTOUT,DOfdata); // sinaliza atuação proteção
    textcolor (LIGHTRED); textbackground(BLACK);
    cprintf("\n\r" );
    cprintf("          FALHA NO BLOCO INVERSOR" );
    switch(qualcomp+VERDADE)
    {
        case VERDADE      : if( (codfal++) == NULL )
            {
                exit(1);
            }
            *codfal = VERDADE+19;
            fwrite ( codfal , sizeof(codfal) , VERDADE , output
            textcolor (LIGHTGREEN); textbackground(BLACK);
            cprintf("\n\r" );
            cprintf("          falha no TBJ 1" );
            break;
/* código falha TBJ 1 */
        case (VERDADE + 1) : if( (codfal++) == NULL )
            {
                exit(1);
            }
            *codfal = VERDADE+20;
            fwrite ( codfal , sizeof(codfal) , VERDADE , output
            textcolor (LIGHTGREEN); textbackground(BLACK);
            cprintf("\n\r" );
            cprintf("          falha no TBJ 2" );
            break;
/* código falha TBJ 2 */
        case (VERDADE + 2) : if( (codfal++) == NULL )
            {
                exit(1);
            }
    }
}
```

```
        *codfal = VERDADE+21 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf("\n\r") ;
        cprintf("          falha no TBJ 3") ;
        break ;
/* código falha TBJ 3 */
    case (VERDADE + 3) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+22 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf("\n\r") ;
        cprintf("          falha no TBJ 4") ;
/* código falha TBJ 4 */
    case (VERDADE + 4) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+23 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf( "\n\r") ;
        cprintf("          falha no TBJ 5") ;
/* código falha TBJ 5 */
    case (VERDADE + 3) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+24 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf( "\n\r") ;
        cprintf("          falha no TBJ 6") ;
/* código falha TBJ 6 */
    }
    case (VERDADE + 4) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+25 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf("\n\r") ;
        cprintf("          falha no TBJ 5") ;
/* código falha TBJ 5 */
    }
    case (VERDADE + 5) : if( (codfal++) == NULL )
```

```

        {
            exit(1) ;
        }
        *codfal = VERDADE+26 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf("\n\r") ;
        cprintf("                falha no TBJ 6" ) ;
/* código falha TBJ 6 */
    }
    if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+27 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf("\n\r" t "" t "Falha de Sobrecarga Contínua(SINAL ANALOGICO)\n\r
/* código falha sobrecarga contínua */
    }

/* parâmetros temperatura TBJ */
fscanf( analog , "%f" , &statusan ) ;
itav = statusan ;
fscanf( analog , "%f" , &statusan ) ;
itrms = statusan ;
fscanf( analog , "%f" , &statusan ) ;
tc = statusan ;

if ( (VTO * itav + RT * ( itrms * itrms ) * ROJC + tc) > tmax )
{
    DOdata = VERDADE ; // falha bloco INVERSOR
    outputb(PORTOUT,DOdata) ; // sinaliza atuação proteção
    textcolor (LIGHTRED) ; textbackground(BLACK) ;
    cprintf( "\n\r" ) ;
    cprintf("                FALHA NO BLOCO INVERSOR") ;
    switch(qualcomp+VERDADE)
    {
        case VERDADE : if( (codfal++) == NULL )
            {
                exit(1) ;
            }
            *codfal = VERDADE+28 ;
            fwrite ( codfal , sizeof(codfal) , VERDADE , output
            textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
            cprintf("\n\r") ;
            cprintf("                falha no TBJ 1" ) ;
            break ;
/* código falha TBJ 1 */
        case (VERDADE + 1) : if( (codfal++) == NULL )

```

```
{
    exit(1);
}
*codfal = VERDADE+29;
fwrite ( codfal , sizeof(codfal) , VERDADE , output-
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r");
cprintf("          falha no TBJ 2");
break;
/* código falha TBJ 2 */
case (VERDADE + 2) : if( (codfal++) == NULL )
    {
        exit(1);
    }
*codfal = VERDADE+30;
fwrite ( codfal , sizeof(codfal) , VERDADE , output
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r");
cprintf("          falha no TBJ 3");
break;
/* código falha TBJ 3 */
case (VERDADE + 3) : if( (codfal++) == NULL )
    {
        exit(1);
    }
*codfal = VERDADE+31;
fwrite ( codfal , sizeof(codfal) , VERDADE , output
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r");
cprintf("          falha no TBJ 4");
/* código falha TBJ 4 */
}
if( (codfal++) == NULL )
    {
        exit(1);
    }
case (VERDADE + 4) : if( (codfal++) == NULL )
    {
        exit(1);
    }
*codfal = VERDADE+32;
fwrite ( codfal , sizeof(codfal) , VERDADE , output
textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r");
cprintf("          falha no TBJ 5");
/* código falha TBJ 5 */
case (VERDADE + 3) : if( (codfal++) == NULL )
    {
        exit(1);
    }
*codfal = VERDADE+33;
```

```

        fwrite ( codfal , sizeof(codfal) , VERDADE , output
        textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf( "\n\r" ) ;
        cprintf("          falha no TBJ 6" ) ;
/* código falha TBJ 6 */
    }
    *codfal = VERDADE+34 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    textcolor (LIGHTGREEN) ; textbackground(BLACK) ;
    cprintf("\n\r" t "" t "Falha de Sobretemperatura(SINAL ANALOGICO)\n\r")
/* código falha sobretemperatura */
}
}
}
}
}

// VERIFICA STATUS PARAM. DIGITAIS SISTEMA - PONTES
INVERSORA/RETIFICADORA/CHAVE ESTÁTICA
if( fscanf( digital , "%s" , binario )!= FALSO )
{
    for( i= FALSO, status= FALSO; binario[i]!='\0'; i++ )
    {
        status<<= VERDADE ;
        status|= (binario[i]=='1') ? VERDADE : FALSO ;
    }
}
else fprintf(stderr, "\n\r\taErro de Leitura em fscanf: Arquivo %s Linha %d",
if ( !BITset( status,MASCARA3) // 6 bits
{
    if(!BITset(status,0x001))
    {
        DOdata = VERDADE ; // falha bloco INVERSOR
        outportb(PORTOUT,DOdata) ; // sinaliza atuação proteção
        if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+35 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
        cprintf("\n\r\n\r" ) ;
        textcolor(YELLOW) ; textbackground(BLACK) ;
        cprintf("          FALHA NO BLOCO INVERSOR" ) ;
        textcolor(LIGHTMAGENTA) ; textbackground(BLACK) ;
        cprintf("\n\r" ) ;
        cprintf("          Falha de Fusível Aberto(SINAL DIGITAL)" ) ;
/* código falha fusível aberto */
    }

    if(!BITset(status,0x002))
    {

```

```
DOdata = VERDADE ; // sinaliza falha bloco INVERSOR
outportb(PORTOUT,DOdata) ;
if( (codfal++) == NULL )
{
    exit(1) ;
}
*codfal = VERDADE+36 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
textcolor(YELLOW) ; textbackground(BLACK) ;
cprintf("\n\r\n\r") ;
cprintf("          FALHA NO BLOCO INVERSOR") ;
cprintf("\n\r") ;
textcolor(LIGHTMAGENTA) ; textbackground(BLACK) ;
cprintf("          Falha de Subtensao(SINAL DIGITAL)") ;
/* código falha subtensao */
}
if(!BITset(status,0x004))
{
    DOdata = VERDADE ; // falha bloco INVERSOR
    outportb(PORTOUT,DOdata) ; // sinaliza atuação proteção
    if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+20 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    textcolor(YELLOW) ; textbackground(BLACK) ;
    cprintf("\n\r\n\r") ;
    cprintf("          FALHA NO BLOCO INVERSOR") ;
    cprintf("\n\r") ;
    textcolor(LIGHTMAGENTA) ; textbackground(BLACK) ;
    cprintf("          Falha de Sobretensão(SINAL DIGITAL)") ;
    /* código falha sobretensão */
}
if(!BITset(status,0x008))
{
    DOdata = VERDADE+3 ; // falha bloco BATERIA
    outportb(PORTOUT,DOdata) ; // sinaliza atuação proteção
    if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+37 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    textcolor(YELLOW) ; textbackground(BLACK) ;
    cprintf("\n\r\n\r") ;
    cprintf("          FALHA NO BLOCO BATERIA") ;
    cprintf("\n\r") ;
    textcolor(LIGHTMAGENTA) ; textbackground(BLACK) ;
```

```
| cprintf("          Falha de Subtensao(SINAL DIGITAL)");  
| /* código falha subtensao bloco Bateria */  
| }  
| if(!BITset(status,0x010))  
| {  
|   DOdata = VERDADE+1 ; //falha bloco RETIFICADOR  
|   outportb(PORTOUT,DOdata) ; // sinaliza atuação proteção  
|   if( (codfal++) == NULL )  
|   {  
|     exit(1) ;  
|   }  
|   *codfal = VERDADE+38 ;  
|   fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;  
|   textcolor(YELLOW) ; textbackground(BLACK) ;  
|   cprintf("\n\r\n\r");  
|   cprintf("          FALHA NO BLOCO RETIFICADOR");  
|   cprintf("\n\r");  
|   textcolor(LIGHTMAGENTA) ; textbackground(BLACK) ;  
|   cprintf("          Falha de Fusível Aberto(SINAL DIGITAL)");  
|   /* código falha fusível aberto bloco Retificador */  
| }  
  
| if(!BITset(status,0x020))  
| {  
|   DOdata = VERDADE+4 ; // falha bloco CHAVE ESTATICA  
|   outportb(PORTOUT,DOdata) ; // sinaliza atuação proteção  
|   if( (codfal++) == NULL )  
|   {  
|     exit(1) ;  
|   }  
|   *codfal = VERDADE+39 ;  
|   fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;  
|   textcolor(YELLOW) ; textbackground(BLACK) ;  
|   cprintf("\n\r\n\r");  
|   cprintf("          FALHA NO BLOCO CHAVE ESTATICA");  
|   cprintf("\n\r");  
|   textcolor(LIGHTMAGENTA) ; textbackground(BLACK) ;  
|   cprintf("          Falha de Fusível Aberto(SINAL DIGITAL)");  
|   /* código falha fusível aberto bloco Chave Estática */  
| }  
| }  
  
| /* VERIFICA STATUS ANALOGICO SISTEMA - PONTE INVERSORA *****  
| fscanf( analog , "%f" , &statusan ) ;  
| if ( statusan > sobreccorcap )  
| {  
|   DOdata = VERDADE ; // falha bloco INVERSOR  
|   outportb(PORTOUT,DOdata) ; // sinaliza atuação proteção  
|   if( (codfal++) == NULL )  
|   {
```

```

    exit(1);
}
*codfal = VERDADE+40;
fwrite ( codfal , sizeof(unsigned) , VERDADE , output );
textcolor(YELLOW); textbackground(BLACK);
cprintf("\n\r\n\r");
cprintf("          FALHA NO BLOCO CAPACITOR");
cprintf("\n\r");
textcolor(LIGHTMAGENTA); textbackground(BLACK);
cprintf("          Falha de Sobrecorrente(SINAL ANALOGICO)");
/* código falha sobrecorrente sobre capacitor */
}
fscanf( analog , "%f" , &statusan );
if ( statusan > sobreccorbat )
{
DOdata = VERDADE+3; // falha bloco inversor
outportb(PORTOUT,DOdata); // sinaliza atuação proteção
if( (codfal++) == NULL )
{
    exit(1);
}
*codfal = VERDADE+40;
fwrite ( codfal , sizeof(unsigned) , VERDADE , output );
textcolor(YELLOW); textbackground(BLACK);
cprintf("\n\r\n\r");
cprintf("          FALHA NO BLOCO BATERIA");
cprintf("\n\r");
textcolor(LIGHTMAGENTA); textbackground(BLACK);
cprintf("          Falha de Sobrecorrente(SINAL ANALOGICO)");
/* código falha sobrecorrente sobre bateria */
}

textcolor(LIGHTGREEN); textbackground(BLACK);
for ( qualcomp = FALSO ; qualcomp <= (VERDADE + 6) ; qualcomp++ )
/* varre 8 SCRs */
{
if( fscanf( digital , "%s" , binario )!= FALSO )
{
for( i= FALSO, status= FALSO; binario[i]!='\0'; i++ )
{
status<<= VERDADE ;
status|= (binario[i]=='1') ? VERDADE : FALSO ;
}
}
else fprintf(stderr, "\n\r\n\r aErro de Leitura em fscanf: Arquivo %s Linha %d",
if ( (qualcomp == FALSO) || (qualcomp == (VERDADE+1)) || (qualcomp ==
(VERDA-
{
status >>= (VERDADE + 3) ;
}

```

```

if ( !BITset(status,MASCARA4) ) //4 bits
{
    if( qualcomp <= (VERDADE + 4) )
    {
        DOdata = (VERDADE+1) ;
        // sinaliza falha bloco RETIFICADOR
        outputb(PORTOUT,DOdata) ; // sinaliza atuação proteção
        if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+41 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
        textcolor(LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf("          FALHA NO BLOCO RETIFICADOR") ;
    }
    else{
        DOdata = (VERDADE + 2);
        // falha bloco CHAVE ESTATICA
        outputb(PORTOUT,DOdata) ; // sinaliza atuação proteção
        if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+42 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
        textcolor(LIGHTGREEN) ; textbackground(BLACK) ;
        cprintf("          FALHA NO BLOCO CHAVE ESTATICA") ;
    }
    switch (qualcomp+VERDADE)
    {
        case (VERDADE) : if( (codfal++) == NULL )
            {
                exit(1) ;
            }
            *codfal = VERDADE+43 ;
            fwrite ( codfal , sizeof(codfal) , VERDADE , out-
            textcolor(LIGHTBLUE) ;
            cprintf("\n\r") ;
            cprintf("          falha no SCR 1") ;
            break ;
        /* código falha SCR 1 */
        case (VERDADE+1) : if( (codfal++) == NULL )
            {
                exit(1) ;
            }
            *codfal = VERDADE+44 ;
            fwrite ( codfal , sizeof(codfal) , VERDADE , o
            textcolor(LIGHTBLUE) ; textbackground(BLAC
            cprintf("\n\r") ;
    }
}

```

```
                cprintf("                falha no
                break ;
/* código falha SCR 2 */
    case (VERDADE+2) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+45 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("                falha no
        break ;
/* código falha SCR 3 */
    case (VERDADE+3) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+46 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("                falha no
        break ;
/* código falha SCR 4 */
    case (VERDADE+4) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+47 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLACK) ;
        cprintf("\n\r") ;
        cprintf("                falha no
        break ;
/* código falha SCR 5 */
    case (VERDADE+5) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+48 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLACK) ;
        cprintf("\n\r") ;
        cprintf("                falha no SCR 6")
        break ;
/* código falha SCR 6 */
    case (VERDADE+6) : if( (codfal++) == NULL )
        {
            exit(1) ;
```

```
    }
    *codfal = VERDADE+49 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , o
    textcolor(LIGHTBLUE) ; textbackground(BLACK) ;
    cprintf("\n\r") ;
    cprintf("                falha no SCR 7")
    break ;
/* código falha SCR 7 */
    case (VERDADE+7) : if( (codfal++) == NULL )
    {
    exit(1) ;
    }
    *codfal = VERDADE+50 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , o
    textcolor(LIGHTBLUE) ; textbackground(BLACK) ;
    cprintf("\n\r") ;
    cprintf("                falha no SCR 8")-
/* código falha SCR 8 */
    }
    if ( !BITset(status,0x001))
    {
    if( (codfal++) == NULL )
    {
    exit(1) ;
    }
    *codfal = VERDADE+51 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    cprintf("\n\r" t "" t "Falha de Fusível Aberto(SINAL DIGITAL)\n\r"
/* código falha fusível aberto */
    }
    if ( !BITset(status,0x002))
    {
    if( (codfal++) == NULL )
    {
    exit(1) ;
    }
    *codfal = VERDADE+52 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    cprintf("\n\r" t "" t "Falha de Ausência de Pulsos(SINAL DIGITAL)\
/* código falha ausência pulsos */
    }
    if ( !BITset(status,0x004))
    {
    if( (codfal++) == NULL )
    {
    exit(1) ;
    }
    *codfal = VERDADE+53 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    cprintf("\n\r" t "" t "Falha no Circuito de Disparo(SINAL DIGITAL)
```

```
/* código falha circuito disparo */
}
if ( !BITset(status,0x008))
{
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+54 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
cprintf("\n\r" t "" t "Falha no Sistema de Refrigeração(SINAL DIGITAL");
/* código falha sistema refrigeração */
}
}
// ***** VERIFICA STATUS PARAM. ANALOGICOS RETIFICADOR - PONTE
RETIFICADORA **
fscanf( analog , "%f" , &statusan ) ;
if ( statusan > condrevmax )
{
if ( qualcomp <= (VERDADE + 4) )
{
DOdata = (VERDADE+1) ; // falha bloco Retificador
outportb(PORTOUT,DOdata) ; // sinaliza atuação proteção
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+55 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
textcolor(LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("\n\r") ;
cprintf("          FALHA NO BLOCO RETIFICADOR") ;
}
else
{
DOdata = VERDADE + 2 ; // falha bloco CHAVE ESTÁTICA
outportb(PORTOUT, DOdata) ; // sinaliza atuação proteção
if( (codfal++) == NULL )
{
exit(1) ;
}
*codfal = VERDADE+56 ;
fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
textcolor(LIGHTGREEN) ; textbackground(BLACK) ;
cprintf("          FALHA NO BLOCO CHAVE ESTATICA") ;
}
}
//
switch (qualcomp+VERDADE)
{
case (VERDADE) : if( (codfal++) == NULL )
```

```
        {
        exit(1) ;
        }
        *codfal = VERDADE+57 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , out
        textcolor(LIGHTBLUE) ; textbackground(BLACK)
        cprintf("\n\r") ;
        cprintf("          falha no SCR 1") ;
        break ;
/* código falha SCR 1 */
        case (VERDADE+1) : if( (codfal++) == NULL )
        {
        exit(1) ;
        }
        *codfal = VERDADE+58 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("          falha no SCR 2")
        break ;
/* código falha SCR 2 */
        case (VERDADE+2) : if( (codfal++) == NULL )
        {
        exit(1) ;
        }
        *codfal = VERDADE+59 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("          falha no SCR 3")
        break ;
/* código falha SCR 3 */
        case (VERDADE+3) : if( (codfal++) == NULL )
        {
        exit(1) ;
        }
        fwrite ( codfal , sizeof(codfal) , VERDADE , o-
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("          falha no SCR 4")
        break ;
/* código falha SCR 4 */
        case (VERDADE+4) : if( (codfal++) == NULL )
        {
        exit(1) ;
        }
        *codfal = VERDADE+60 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
```

```
        cprintf("                falha no SCR 5")
        break ;
/* código falha SCR 5 */
    case (VERDADE+5) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+61 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("                falha no SCR 6")
        break ;
/* código falha SCR 6 */
    case (VERDADE+6) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+62 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("                falha no SCR 7")
        break ;
/* código falha SCR 7 */
    case (VERDADE+7) : if( (codfal++) == NULL )
        {
            exit(1) ;
        }
        *codfal = VERDADE+63 ;
        fwrite ( codfal , sizeof(codfal) , VERDADE , o
        textcolor(LIGHTBLUE) ; textbackground(BLAC
        cprintf("\n\r") ;
        cprintf("                falha no SCR 8")
/* código falha SCR 8 */
    }
    if( (codfal++) == NULL )
    {
        exit(1) ;
    }
    *codfal = VERDADE+64 ;
    fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
    cprintf("\n\r" t "" t "Falha de Condução Reversa(SINAL ANALOGICO))
/* código falha condução reversa */
}
}
fscanf( analog , "%f" , &statusan ) ;
if ( statusan > ilmax )
{
    deltat = TLIMITE1 ;
```

```

delay( deltat ) ;
// proporciona atraso (sobrecor.) deltaT milissegundos
fscanf( analog , "%f" , &statusan ) ;
if ( statusan > ilmax ) /* verifica sobrecorrentes consecutivas */
{
  DOdata = (VERDADE+1) ; // falha bloco RETIFICADOR
  outportb(PORTOUT,DOdata) ; // sinaliza atuação proteçao
  if( (codfal++) == NULL )
  {
    exit(1) ;
  }
  *codfal = VERDADE+65 ;
  fwrite ( codfal , sizeof(codfal) , VERDADE , output ) ;
  textcolor(LIGHTGREEN) ; textbackground(BLACK) ;
  cprintf("\n\r") ;
  cprintf("          FALHA NO BLOCO RETIFICADOR") ;
  textcolor(LIGHTBLUE) ; textbackground(BLACK) ;
  cprintf("\n\r") ;
  cprintf(" t " t "Falha de Sobrecorrentes Sucessivas(SINAL ANALOGICO)
}
}

textcolor(YELLOW+BLINK) ; textbackground(BLACK) ;
cprintf("\n\r\n\r") ;
window(VERDADE, 24+VERDADE, VERDADE+79, 24+VERDADE) ;
cprintf("Pressione tecla <ESC> para terminar.") ;

Executar = !( feof(digital) || feof(analog) ) ;

if ( kbhit() )
{
  if( ( tecla= getch() ) == ESC )
  {
    Executar= FALSO;
  }
}

} while( Executar == VERDADE ) ;

fclose(digital) ; // fecha todos arquivos dados
fclose(analog) ;
fclose(output) ;
}

//***** UFA!!! *****

```





## ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

### APÊNDICE B



1111111111111111	8
1111111111111111	9
1111111111111111	10
1111111111111111	11
1111111111111111	12
1111111111111111	13
1111111111111110	125
1111111111111111	126
1111111111110111	122
1111111111111111	0
1111111111111111	1
(195 linhas)	(205 linhas)

### B.3 – Todas Falhas

Digital – TODAFALD.DAT

Analógico – TODAFALA.DAT

1111111111100000	122
1111111111111111	122
11111111111000111	122
1111111111111000	122
11111111111000111	122
1111111111111000	122
11111111111000000	122
11111111100001111	122
1111111111110000	122
1111111000011111	122
1111111111110000	122
11111111100001111	122
1111111111110000	122
11111111100001111	122
1111111111110000	122
1111111111100000	122
1111111111111111	122
11111111111000111	122
1111111111111000	122
11111111111000111	122
.....	.....
(240 linhas)	(340 linhas)

## APÊNDICE C – TEMPOS DE EXECUÇÃO DOS BLOCOS DA UPS

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,044626	0,040764	0,044197
0,010941	0,008367	0,003003
0,03776	0,009439	0,003217
0,010727	0,03304	0,002788
0,032611	0,029607	0,00278
0,010727	0,030036	0,002788
0,010731	0,008796	0,002574
0,011156	0,010298	0,021025
0,10512	0,010727	0,002788
0,01156	0,009869	0,003003

Tabela 9 – Hardware 80486 Bloco Start-Up

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,037331	0,051492	0,169066
0,036688	0,040764	0,196744
0,036688	0,04291	0,17915
0,035829	0,041837	0,18194
0,036902	0,043124	0,196744
0,035615	0,042266	0,204682
0,003647	0,041193	0,218843
0,036044	0,0422051	0,176147
0,003647	0,041408	0,191165
0,035615	0,041622	0,188591

Tabela 10 – Hardware 80486 Bloco Inversor

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,01652	0,038833	0,289859
0,019523	0,00429	0,235792
0,020382	0,025102	0,244589
0,019523	0,026175	0,267761
0,020596	0,02614	0,245662
0,019523	0,026389	0,241156
0,021025	0,009654	0,241156
0,019738	0,026604	0,220774
0,020811	0,026604	0,261753
0,019738	0,02596	0,228712

Tabela 11 – Hardware 80486 Bloco Sistema

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,053423	0,05235	0,289859
0,047201	0,01695	0,235792
0,04763	0,04355	0,244589
0,047201	0,03368	0,267761
0,04715	0,04141	0,245662
0,047201	0,00987	0,241156
0,04763	0,02639	0,241156
0,0472	0,01566	0,220774
0,047415	0,01566	0,261753
0,047419	0,02532	0,228712

Tabela 12 – Hardware 80486 Bloco Retificador/Chave Estática

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,039262	0,01505	0,045484
0,026389	0,04763	0,04312
0,031109	0,035829	0,022742
0,007938	0,032826	0,040335
0,033469	0,033255	0,031324
0,043339	0,052779	0,023814
0,044626	0,050204	0,048059
0,006436	0,004505	0,019094
0,043982	0,027891	0,049561
0,015018	0,046342	0,02313

Tabela 13 – Hardware Pentium Bloco Start – Up

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,036259	0,03733	0,198246
0,042695	0,05042	0,195456
0,015876	0,02489	0,20833
0,031324	0,02489	0,202537
0,006221	0,01373	0,213479
0,43553	0,03969	0,219486
0,001286	0,0266	0,202322
0,038404	0,01373	0,209402
0,037546	0,03604	0,131949
0,034328	0,0442	0,217984

Tabela 14 – Hardware Pentium Bloco Inversor

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,013516	0,01008	0,06887
0,035615	0,03326	0,09805
0,019309	0,003	0,043553
0,032182	0,0221	0,105344
0,032611	0,02446	0,10277
0,038189	0,02274	0,076594
0,003217	0,01352	0,109635
0,001501	0,01352	0,108777
0,006436	0,01073	0,061361
0,046342	0,02381	0,088609

Tabela 15- Hardware Pentium Bloco Sistema

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,025316	0,06308	0,226567
0,041622	0,04892	0,23944
0,016305	0,03068	0,209617
0,037331	0,06608	0,184943
0,000428	0,06458	0,196958
0,030251	0,04205	0,210261
0,01416	0,06437	0,228283
0,035186	0,03948	0,274412
0,025531	0,05707	0,225923
0,025102	0,06758	0,210261

Tabela 16 – Hardware Pentium Bloco Retificador/Chave Estática

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,007938	0,040979	0,033255
0,027891	0,009869	0,021669
0,025102	0,01137	0,006007
0,036902	0,006436	0,003217
0,030466	0,019738	0,043339
0,038189	0,032826	0,048273
0,002788	0,016305	0,002574
0,02832	0,010512	0,044412
0,04999	0,048703	0,002574
0,022956	0,008367	0,0118

Tabela 17 – Hardware Pentium II Bloco Start – Up

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,043768	0,049561	0,193955
0,001501	0,031753	0,183227
0,03304	0,031324	0,184085
0,028964	0,011585	0,217341
0,00193	0,04291	0,185587
0,015232	0,045913	0,184729
0,003003	0,038404	0,163917
0,022742	0,041193	0,186874
0,002788	0,041204	0,196315
0,022956	0,041837	0,182154

Tabela 18 – Hardware Pentium II Bloco Inversor

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,042266	0,023171	0,074449
0,015876	0,0236	0,073805
0,04012	0,037975	0,046986
0,039477	0,042481	0,058357
0,044841	0,009654	0,055354
0,026175	0,035615	0,079384
0,034971	0,011585	0,096333
0,023814	0,003432	0,071874
0,053208	0,022742	0,086893
0,013301	0,00901	0,025316

Tabela 19 – Hardware Pentium II Bloco Sistema

NÚMERO DE FALHAS		
Nenhuma	Uma	Todas
0,02832	0,058143	0,259393
0,030036	0,027247	0,240727
0,046986	0,026818	0,237723
0,012658	0,032182	0,24589
0,044412	0,052779	0,202751
0,026604	0,059001	0,242443
0,053208	0,080671	0,196958
0,037331	0,008367	0,244374
0,024458	0,003003	0,264328
0,026604	0,056212	0,222705

Tabela 20 – Hardware Pentium II Bloco Retificador/Chave Estática



Escola Federal de Engenharia de Itajubá

## APÊNDICE D

## Apêndice D – EXEMPLOS DE TELAS DE VISUALIZAÇÃO DE FALHAS

### FALHA NO BLOCO START-UP

Falha de Alimentação nos Drives(SINAL DIGITAL)  
 Falha de Subtensão(SINAL DIGITAL)  
 Falha de Ausência de Fases(SINAL DIGITAL)  
 Falha na Seqüência de Fases(SINAL DIGITAL)  
 Falha no Sistema de Refrigeração(SINAL DIGITAL)

Quadro 3 – Falhas no Bloco Start-Up

### FALHA NO BLOCO INVERSOR

#### Falha no TBJ 3

Falha de Fusível Aberto(SINAL DIGITAL)  
 Falha de Ausência de Pulsos(SINAL DIGITAL)  
 Falha no Sistema de Refrigeração(SINAL DIGITAL)

### FALHA NO BLOCO INVERSOR

#### Falha no TBJ 3

Falha de Sobrecorrente(SINAL ANALÓGICO)

### FALHA NO BLOCO INVERSOR

#### Falha no TBJ 6

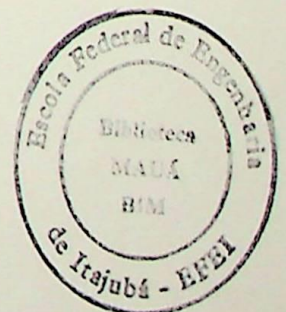
Falha de Fusível Aberto(SINAL DIGITAL)  
 Falha de Ausência de Pulsos(SINAL DIGITAL)  
 Falha no Sistema de Refrigeração(SINAL DIGITAL)

### FALHA NO BLOCO INVERSOR

#### Falha no TBJ 6

Falha de Sobrecorrente(SINAL ANALÓGICO)

Quadro 4 - Falhas no Bloco Inversor



**FALHA NO BLOCO INVERSOR****Falha de Fusível Aberto(SINAL DIGITAL)****FALHA NO BLOCO INVERSOR****Falha de Subtensão(SINAL DIGITAL)****FALHA NO BLOCO INVERSOR****Falha de Sobretenção(SINAL DIGITAL)****FALHA NO BLOCO BATERIA****Falha de Subtensão(SINAL DIGITAL)****FALHA NO BLOCO BATERIA****Falha de Subtensão(SINAL DIGITAL)****FALHA NO BLOCO RETIFICADOR****Falha de Fusível Aberto(SINAL DIGITAL)****FALHA NO BLOCO CHAVE ESTÁTICA****Falha de Fusível Aberto(SINAL DIGITAL)****FALHA NO BLOCO CAPACITOR****Falha de Sobrecorrente(SINAL ANALÓGICO)****FALHA NO BLOCO BATERIA****Falha de Fusível Aberto(SINAL ANALÓGICO)**

Quadro 5 – Falhas de Sistema

**FALHA NO BLOCO RETIFICADOR****Falha no SCR 4****Falha de Ausência de Pulsos(SINAL DIGITAL)****Falha no Circuito de Disparo(SINAL DIGITAL)****Falha no Sistema de Refrigeração(SINAL DIGITAL)****FALHA NO BLOCO RETIFICADOR****Falha no SCR 4****Falha de Condução Reversa(SINAL ANALÓGICO)****FALHA NO BLOCO CHAVE ESTÁTICA****Falha no SCR 8****Falha de Ausência de Pulsos(SINAL DIGITAL)****Falha no Circuito de Disparo(SINAL DIGITAL)****Falha no Sistema de Refrigeração(SINAL DIGITAL)****FALHA NO BLOCO CHAVE ESTÁTICA****Falha no SCR 8****Falha de Condução Reversa(SINAL ANALÓGICO)****FALHA NO BLOCO RETIFICADOR****Falha de Sobrecorrentes Sucessivas(SINAL ANALÓGICO)**

Quadro 6 – Falhas do Blocos Retificador/Chave Estática



## APÊNDICE E

## Apêndice E – DIAGRAMA EM BLOCOS DE UM UPS

A figura 63 mostra o diagrama em blocos de um UPS típico. O bloco RF é correspondente a uma ponte do tipo semi-controlada, que além de a função de converter a tensão CA da rede de suprimento para tensão DC, possui a função adicional de controlar a carga do banco de baterias. Deve ser observado que as diferentes opções de controle da bateria são implementadas no regulador de tensão e no regulador/limitador de corrente, associados ao módulo de disparo PHS, o qual libera pulsos de gatilhamento para os SCR's da ponte RF.

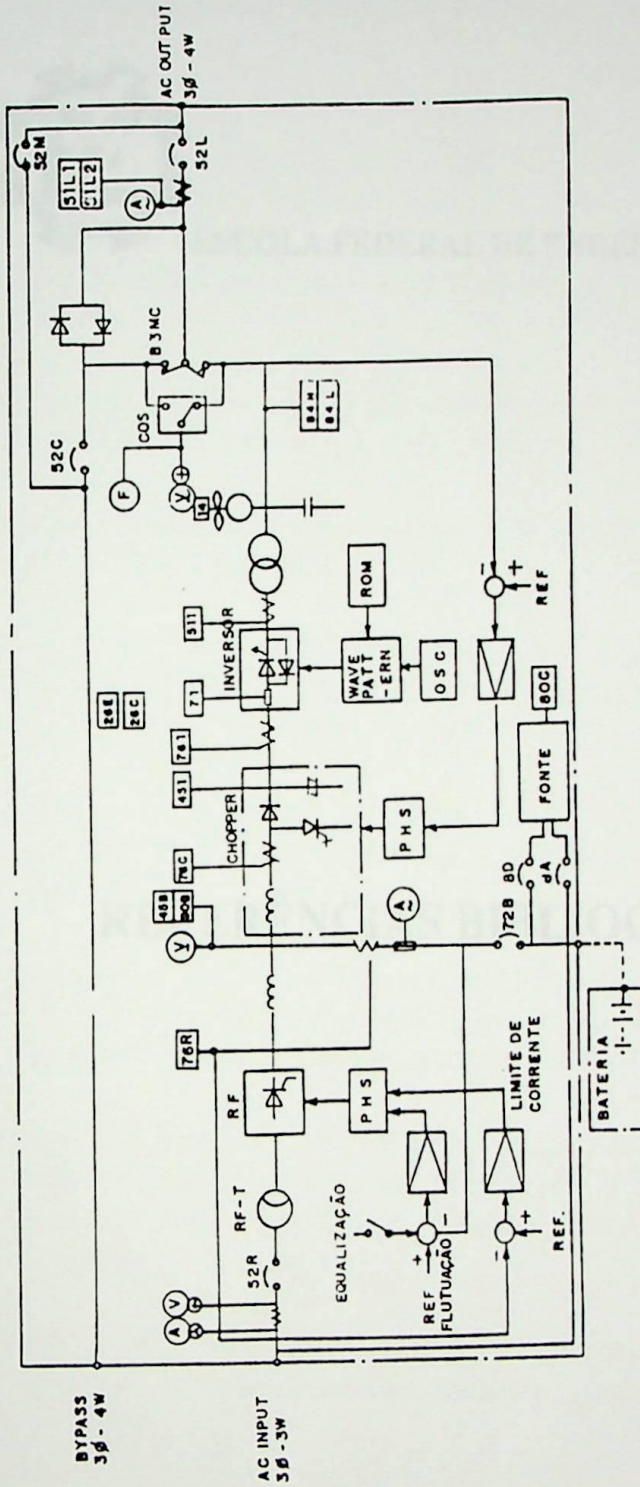
O bloco CHOP é um chopper (conversor DC-DC), cuja finalidade é controlar o nível de tensão DC aplicado à unidade do bloco INV. Este último bloco é um inversor do tipo tensão imposta (VSI), cuja finalidade é transformar a tensão de contínua em uma tensão alternada.

Observa-se ainda um transformador, que é utilizado para adequar a tensão de saída do inversor com a tensão da carga, e um filtro cuja finalidade é eliminar as harmônicas presentes no sinal de tensão do transformador.

Nota-se no diagrama de blocos, que existe uma malha de controle da tensão de saída, cujo regulador controla o bloco PHS, correspondente ao controle do chopper. Também observa-se que existe uma memória ROM, que contém todos os tempos de chaveamento ON/OFF correspondentes aos gatilhamentos dos semicondutores de potência (IGBT, GTO ou PBT) do módulo INV.

A chave estática de transferência é constituída de dois SCRs em anti-paralelo e um contator.

Figura 63 – Diagrama em Blocos Típico de um UPS



- PHS: PHASE SHITLER
- OSC: Oscilador
- COS: Chave de Transferência para Medidor
- ROM: Memória ROM
- WAVE: Padrão de Pulsos Armazenados

Figura 63 – Diagrama em Blocos Típico de um UPS

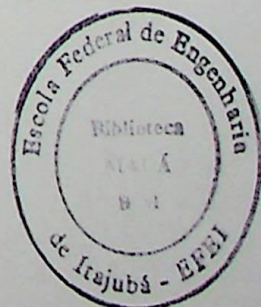


ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

## REFERÊNCIAS BIBLIOGRÁFICAS

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Reeve, J. , Logic behaviour of H.V.D.C. converters during normal and abnormal conditions . **Proc. IEE** vol 114, n. 12, p. 1937 – 46, 1967.
- [2] Silva, V. F., **Eletrônica Industrial II**, D. A.E.F.E.I., Itajubá, vol. I, 1ª edição, 150 p., 1985.
- [3] Arrillaga, J. , Galanos, G., Fault – development control in A. C.-D.C.converters , **Proc. IEE**, vol. 116, n. 7, p. 1201 – 08, 1969.
- [4] Reeve, J. , Direct digital protection of H.V.D.C. converters , **Proc. IEE** vol 114, n.12, p. 1947 – 54, 1967.
- [5] Arrillaga, J. ; Galanos, G., Theorical basis of a digital method of grid control of H.V.D.C. converters , **IEEE Transactions and Power Apparatus and System**, vol. PAS-89, n. 8, p. 2049 – 55, 1970.
- [6] Arrillaga, J. ; Galanos, G.; Towner, E. T., Direct digital control of H.V.D.C. converters, **IEEE Transactions and Power Apparatus and System**, vol. PAS-89, n. 8, p. 2056 – 65, 1970.
- [7] Ranade. S.; Rumpf, E., Comparison of suitable control systems for H.V.D.C. stations connected to weak A.C. systems – part 1: New control systems, **Transactions and Power Apparatus and Systems**, p. 549 –54, 1971.
- [8] Ranade, S.; Rumpf, E., Comparison of suitable control systems for H.V.D.C. stations connected to weak A.C. systems – part 2: Operational behaviour of the H.V.D.C. transmission, **Transactions and Power Apparatus and Systems**, p. 555 – 64, 1971.
- [9] Peppel, J. M.; Salzmann, T., GTO Driving and protection technique with status monitoring, **Transactions on Industry Applications**, vol 24, no 1, p. 115 – 20, 1988.
- [10] Dubey, G. K.; Murty, Y. V. V.; Sinha, R. M. K., Fault diagnosis in three – phase thyristor converters using microprocessor. **IEEE Transactions on Industry Applications** vol IA-20, n 6, p. 1940 – 47, 1984.
- [11] Cortez, J. A.; Aderbal A. P. Jr; Jocélio S. de Sá; Angelo J. J. Rezek. - “**Bancada de Ensaio para Avaliar o Comportamento de Acionamentos Controlados por Inversor PWM**” - II Seminário de Estudos de Engenharia Elétrica , UFU, Uberlândia MG, Março de 1998.



- [12] Biswas, S. K.; Basak, B.; Rajashekara K. S., A modular gate drive circuit for insulated gate bipolar transistors, **IEEE Transactions of Industry Applications**, vol. 24, n<sup>o</sup>. 1, p. 1490 - 95, 1991.
- [13] Schildt, H., **Turbo C - Guia do Usuário**, McGraw - Hill, São Paulo. 2<sup>a</sup> edição, 414 p., 1988.
- [14] Mendonça A ., Zelenovsky R., **PC e Periféricos**, Editora Ciência Moderna, Rio de Janeiro, 493 p., 1996.
- [15] Schildt, H., **Linguagem C - Guia Prático e Interativo**, McGraw - Hill, São Paulo, 2<sup>a</sup> edição, 363 p., 1988.
- [16] Schildt, H., **C Avançado**, McGraw - Hill, São Paulo, 2<sup>a</sup> edição, 335 p., 1987.
- [17] Cabral, F., **a Linguagem C e o PC – BIOS**, Editora Campus, Rio de Janeiro, 142 p., 1989.
- [18] **Motorola CMOS Logic Data**, Motorola Incorporated, Austin, TX, 590 p., 1985.
- [19] Labtech Acquire. **Laboratory Technologies Corporation**. Wilmington MA, 1988.
- [20] Vermaas, L. L. G.. **Implementação de um Sistema de Controle Digital para Conversor Dual baseado em Microcontrolador**. Dissertação(Mestrado em Automação Industrial), Escola Federal de Engenharia de Itajubá. 107 p., 1995.