

UNIVERSIDADE FEDERAL DO PARANÁ

EDUARDO LUIZ SOPPA

SISTEMA TOLERANTE A FALHAS UTILIZANDO WINDOWS CE,  
FOCANDO CONTROLADORES INDUSTRIAIS

CURITIBA

2009

EDUARDO LUIZ SOPPA

SISTEMA TOLERANTE A FALHAS UTILIZANDO WINDOWS CE,  
FOCANDO CONTROLADORES INDUSTRIAIS

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre em Engenharia  
Elétrica, Programa de Pós-Graduação em  
Engenharia Elétrica, Departamento de  
Engenharia Elétrica, Setor de Tecnologia,  
Universidade Federal do Paraná.

Orientador: Prof. José Manoel Fernandes, Ph.D.

CURITIBA

2009

## TERMO DE APROVAÇÃO

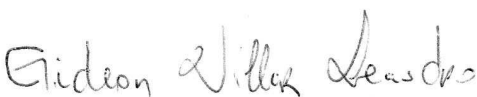
EDUARDO LUIZ SOPPA

### SISTEMA TOLERANTE A FALHAS UTILIZANDO WINDOWS CE, FOCANDO CONTROLADORES INDUSTRIAIS

Dissertação aprovada como requisito parcial para a obtenção do grau de Mestre no Curso de Pos-Graduação em Engenharia Elétrica, Setor de Tecnologia da Universidade Federal do Paraná, pela seguinte banda examinadora:

  
Orientador: Prof. José Manoel Fernandes, Ph.D.  
Departamento de Engenharia Elétrica, UFPR.

  
Prof. Leandro dos Santos Coelho, Dr.  
Departamento de Engenharia Elétrica, UFPR.

  
Prof. Gideon Villar Leandro, Dr.  
Departamento de Engenharia Elétrica, UFPR.

  
Prof. Eduardo de Freitas Rocha Loures, Dr.  
Departamento de Engenharia Mecatrônica, PUC-PR.

Curitiba, 07 de julho de 2009.

*Ao meu querido e amado filho, João Paulo,  
que todos os dias me motiva com seus convites irrecusáveis.*

*- Papai, vamos brincar?*

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus, pois sem Ele nada é possível.

Ao amigo e Prof. José Manoel Fernandes, Ph.D. por todo apoio, conselhos, orientações e por acreditar neste trabalho.

Aos professores do programa de Pós-Graduação em Engenharia Elétrica da UFPR pelo esforço constante em oferecer um ensino de qualidade e aos professores que aceitaram fazer parte desta banca.

Agradeço aos meus pais Luiz Soppa e Rejansira Soppa por todo amor, carinho, apoio incondicional e por sempre estarem ao meu lado.

A minha amada esposa Alexandra C. Guimarães, pela dedicação, apoio, carinho e pelas palavras de incentivo.

Ao meu filho João Paulo por me acordar com beijos nos sábados e nos domingos e a Maria Tereza pelos seus sorrisos, verdadeira fonte de alegria.

As minhas irmãs Luciane e Daniele por me apoiarem em vários momentos.

Ao amigo Eng. Juliano de Mello Pedroso, M.Sc. pelas sugestões e contribuições feitas a este trabalho. Que as experiências compartilhadas até aqui sejam a alavanca para alcançarmos à alegria de chegar ao destino por cada um de nós projetado.

Aos amigos do Senai, Marcio Debner, Carlos Sakiti e Marco Túlio pelos incontáveis debates na padaria na hora do intervalo, e a Miguel Igino por confiar em meu trabalho.

A todos aqueles que diretamente ou indiretamente contribuíram para a realização desta dissertação. E, a todos aqueles que entenderam a minha ausência durante este período.

Ao Senai-CIC pela infra-estrutura cedida para a realização deste estudo.

Se sou fiel no pouco, Ele me confiará mais.  
Se sou fiel no pouco, meus passos guiará.

Lucas (19, 11-28)

## RESUMO

Os Sistemas Instrumentados de Segurança (SISs) são sistemas responsáveis pela segurança operacional de unidades e equipamentos industriais. Eles causam a parada de emergência (ESD – *Emergency Shutdown*) ou impedem uma operação insegura sempre que as condições do processo ultrapassem os limites pré-estabelecidos como seguros. Após alguns acidentes fatais, com perda de vidas humanas, agressão do meio ambiente, começou-se a por em prática o conhecimento em segurança e o desenvolvimento de sistemas. Isso refletiu em normas nacionais (N-2595) e internacionais (IEC-61508, IEC-61511, TR-84.01) no desenvolvimento de produtos específicos para serem usados na concepção e arquitetura de novos projetos de instalações. Existem diferentes tipos de sistemas (arquitetura) que atendem a diferentes processos. Muitas vezes, após ter sido feita a qualificação da malha de controle, o preço é quem decide qual sistema será implantado, transformando o custo do sistema em um item de decisão, desde que os sistemas analisados atendam o mesmo nível de integridade de segurança (SIL - *Safety Integrity Level*). O objetivo desta dissertação é propor e validar de forma prática um Sistema Redundante de Segurança utilizando o *hardware* de um PC (*Personal Computer*) padrão, uma vez que este *hardware* já está bem testado. Ainda, demonstrar que é possível o desenvolvimento de SIS de baixo custo, tão eficiente quanto os sistemas existentes no mercado. A diferença entre o sistema proposto e um computador de uso comum será o sistema operacional, pois neste projeto irá ser utilizado um sistema embarcado (Windows CE) no lugar dos sistemas operacionais conhecidos, de maneira que os erros provenientes do *software* que ocorrem nos computadores sejam eliminados. Para que se tenha a redundância modular tripla (TMR – *Triple Modular Redundant*) irá ser colocado três computadores com sistema embarcado para processar as informações em paralelo (ao mesmo tempo). Para que a interface com o mundo externo (sensores, transmissores e atuadores) seja feita, placas de comunicação se farão necessárias, conversores de corrente para tensão, conversor de pulso PWM (*Pulse Width Modulation*) para sinal analógico e modulação em corrente etc. Todo o sistema será testado em uma planta didática e deverá efetuar o controle de vazão de uma malha, utilizando para tanto um algoritmo PID (Proporcional Integral e Derivativo) devidamente sintonizado. Ainda, será identificada a probabilidade de falha sobre demanda (PFD) para que seja possível a qualificação do sistema de acordo com a norma IEC 61508.

**Palavras-chave:** Sistema Instrumentado de Segurança (SIS). Nível de Integridade de Segurança (SIL – *Safety Integrity Level*). Sistema embarcado. Controle. PID (Proporcional Integral e Derivativo). Probabilidade de Falha sobre Demanda (PFD). IEC 61508. Redundância Modular Tripla (TMR - *Triple Modular Redundant*). Paralelo. Intertravamento. Windows CE.

## ABSTRACT

The Safety Instrumented Systems (SISs) are systems responsible for the industrial units and equipments operational safety. They cause emergency stops (ESD – Emergency Shutdown) or prevent unsafe actions whenever the process conditions go outside pre-established safety limits. After several fatal accidents, like loss of human lives and environmental harm that the safety knowledge and systems design was put in practice resulting in national (N-2595) and international (IEC-61508, IEC-61511, TR-84.01) norms to be used in specific products for concept and design of new installation projects. Many times after loop qualification its price influences which system will be implemented, making the system's cost in a decision item, since all analyzed systems meet the same Safety Integrity Level (SIL). The objective of this dissertation is to propose and validate in a practical way a redundant safety system using a standard PC (Personal Computer) hardware. As this hardware is widely tested and proven to work, it can be demonstrated that a low cost SIS can be developed as effective as other existing market SIS. A difference between the proposed system and one common use computer will be the operating system, because an embedded system (Windows CE) instead of known operating systems, can debug known errors. To obtain a Triple Modular Redundancy (TMR) three computers with embedded system will process the information in parallel (at the same time). For external interface (sensors, transmitters and actuators) communication boards will be developed, current to voltage converters, pulse PWM (Pulse Width Modulation) for analogical signals, current modulation among others. All this system will be employed in a didactic plant and must execute flow control in a loop using a PID (Proportional Integral Derivative) algorithm. And the, failure probability over demand (PFD) will still be identified, so it will be possible to qualify the system in accordance with the norm IEC61508.

**Keywords:** Sistema Instrumentado de Segurança (SIS). *Safety Integrity Level* (SIL). Embedded System. Control. Proportional Integral Derivative (PID). Failure Probability over Demand (PFD). IEC 61508. Triple Modular Redundancy (TMR). Parallel. Interlock. Windows CE.



## LISTA DE ILUSTRAÇÕES

FIGURA 1 – ETAPAS METODOLÓGICAS.....	19
FIGURA 2 – ALARP.....	26
FIGURA 3 – CAUSA COMUM.....	39
FIGURA 4 – SISTEMA DE REDUNDÂNCIA MODULAR TRIPLA.....	42
FIGURA 5 – SISTEMA <i>HOT STANDBY</i> .....	43
FIGURA 6 – BLOCOS DE RECUPERAÇÃO.....	44
FIGURA 7 – O IMPACTO DA REDUNDÂNCIA.....	45
FIGURA 8 – SISTEMA DE CONTROLE.....	52
FIGURA 9 – SISTEMA A MALHA ABERTA.....	52
FIGURA 10 – SISTEMA A MALHA FECHADA.....	53
FIGURA 11 – SISTEMA DE CONTROLE CLÁSSICO.....	55
FIGURA 12 – SISTEMA DE PARADA PNEUMÁTICO.....	58
FIGURA 13 – SISTEMA A RELE TÍPICO.....	60
FIGURA 14 – CONCEITO DE OPERAÇÃO.....	61
FIGURA 15 – SISTEMA 2oo2 COM DIAGNÓSTICO.....	64
FIGURA 16 – DIAGRAMA DE BLOCOS DE UM CANAL DO SISTEMA PROPOSTO. .....	65
FIGURA 17 – DIAGRAMA DE INTERLIGAÇÃO.....	66
FIGURA 18 – DIAGRAMA DA PLANTA 3.....	68
FIGURA 19 – DIAGRAMA PROPOSTO.....	70
FIGURA 20 – MODOS DE FALHA DA ARQUITETURA 2oo3.....	72
FIGURA 21 – ÁRVORE DE FALHA DE UMA ARQUITETURA 2oo3 PARA OS CANALIS A e B.....	72
FIGURA 22 – FLUXOGRAMA DO ALGORITMO DO CANAL DE AQUISIÇÃO.....	75
FIGURA 23 – FLUXOGRAMA DA FUNÇÃO DE TRATAMENTO DA INTERRUPÇÃO SERIAL 1.....	76
FIGURA 24 - FLUXOGRAMA DA FUNÇÃO DE TRATAMENTO DA INTERRUPÇÃO SERIAL 2.....	77
FIGURA 25 – FLUXOGRAMA DO ALGORITMO DE CONTROLE DO CANAL DE VOTAÇÃO.....	79

FIGURA 26 – FLUXOGRAMA DO ALGORITMO DE <i>SHUTDOWN</i> DO CANAL DE VOTAÇÃO.....	80
FIGURA 27 – FLUXOGRAMA DA TROCA DE DADOS ENTRE O CANAL DE AQUISIÇÃO E O WINDOWS CE. ....	85
FIGURA 28 – FLUXOGRAMA DO SISTEMA DE CONTROLE.....	86
FIGURA 29 – CURVA DE RESPOSTA DO SISTEMA EM MALHA ABERTA. ....	87
FIGURA 30 – CURVA DE RESPOSTA DO PROCESSO CONTROLADO PELO SISTEMA.....	89
FIGURA 31 – CURVA DE RESPOSTA DO PROCESSO CONTROLADO POR UM CONTROLADOR INDUSTRIAL. ....	89
FIGURA 32 – DIAGRAMA ESQUEMÁTICO DO SISTEMA 2oo3. ....	91
FIGURA 33 – FLUXOGRAMA DO SISTEMA DE INTERTRAVAMENTO. ....	93

## LISTA DE TABELAS

TABELA 1 – NÍVEL DE INTEGRIDADE DE SEGURANÇA IEC 61508 VERSUS DIN/VDE 19250.....	23
TABELA 2 - SIL EM FUNÇÃO DE DISPONIBILIDADE E PROBABILIDADE DE FALHA SOB DEMANDA .....	33
TABELA 3 – CÁLCULO DE $PFD$ E $MTTF_{sp}$ PARA SISTEMAS REDUNDANTES ....	35
TABELA 4 – TABELA DE CLPs DE SEGURANÇA CERTIFICADOS IEC 61508 .....	63
TABELA 5 – EQUIPAMENTOS E SUAS FUNÇÕES. ....	68
TABELA 6 – CONJUNTO DE CARACTERÍSTICAS UTILIZADAS PARA A CONSTRUÇÃO DA IMAGEM NO WINDOWS CE. ....	82
TABELA 7 – TABELA VERDADE DE SAÍDA DE CADA CANAL DE PROCESSAMENTO DADA AS ENTRADAS (SA,SB e SC).....	91
TABELA 8 – FALHAS SEGURAS .....	94
TABELA 9 – FALHAS PERIGOSAS.....	95

## LISTA DE SIGLAS

AIChE	- Instituto Americano de Engenheiros Químicos
ALARP	- <i>As Low as Reasonably Possible</i>
ANSI	- <i>American National Standards Institute</i>
API	- <i>American Petroleum Institute</i>
CCPS	- <i>Center for Chemical Process Safety</i>
CE	- <i>Compact Edition</i>
CLP	- Controlador Lógico Programável
CO	- Saída de Controle
CPU	- Unidade Central de Processamento
DIN	- <i>Deutsches Institut für Normung</i>
EPA	- <i>Environment Program Agency</i>
ESD	- <i>Emergency Shutdown</i>
FMEA	- Análise de Modos e Falhas e seus Efeitos
HAZOP	- <i>Hazard Operability</i>
IEC	- <i>International Electrotechnical Commission</i>
ISA	- <i>Instrumentation and Automation Society</i>
LED	- Diodo Emissor de Luz
LCD	- <i>Liquid Crystal Display</i>
MMU	- <i>Memory Management Unit</i>
MTBF	- Tempo Médio entre Falhas
MTTF	- Tempo Médio de Falha
MTTR	- Tempo Médio de Reparo
MV	- Variável Manipulada
NFPA	- <i>National Fire Protection Agency</i>
NMR	- Redundância Modular N
OMAC	- <i>Open, Modular, Architecture Control</i>
OSHA	- <i>Occupational Safety and Health Organization</i>
PC	- <i>Personal Computer</i>
PES	- Sistema Eletrônico Programável
PFD	- Probabilidade de Falha sobre Demanda
PHA	- <i>Process Hazard Analysis</i>

PID	- Proporcional, Integral e Derivativo
PSM	- <i>Risc Management Program</i>
PV	- Variável do Processo
PWM	- <i>Pulse Width Modulation</i>
RRF	- Fator de Redução de Risco
SAFE	- <i>Safety Function Evaluation</i>
SDCD	- Sistema Digital de Controle Distribuído
SIL	- Nível de Integridade de Segurança
SIS	- Sistema de Intertravamento de Segurança
SP	- <i>Set Point</i>
TMR	- Redundância Modular Tripla
TR	- <i>Technical Report</i>
TUV	- <i>Technische Überwachungs Verein</i>
USB	- <i>Universal Serial Bus</i>
WDT	- <i>Watchdog Timer</i>
Win API	- <i>Application Programming Interface</i>

## SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>15</b>
1.1. OBJETIVOS GERAIS.....	17
1.2. OBJETIVOS ESPECÍFICOS.....	17
1.3. ESTRUTURA DA DISSERTAÇÃO.....	18
1.4. ETAPAS METODOLÓGICAS.....	18
<b>2. REVISÃO DA LITERATURA</b> .....	<b>20</b>
2.1. OS SISTEMAS DE SEGURANÇA NOS PROCESSOS INDUSTRIAIS.....	20
2.2. NORMAS ATUAIS PARA SIS.....	21
2.3. ANÁLISE DO RISCO.....	23
2.3.1. Perigo ( <i>Hazard</i> ).....	23
2.3.2. Risco.....	24
2.3.3. Conceitos de Risco.....	25
2.4. DEFINIÇÕES SOBRE SISTEMAS DE SEGURANÇA (SIS).....	27
2.4.1. Falhas.....	29
2.4.2. Análise Quantitativa e Qualitativa.....	31
2.4.3. Probabilidade de Falha sobre Demanda (PFD).....	32
2.4.4. Análise de Sistemas.....	35
2.4.4.1. Análise de um Sistema a Rele.....	35
2.4.4.2. Análise de um Sistema com Controlador Lógico Programável não redundante.....	36
2.4.4.3. Análise de um Sistema de Redundância Tripla (TMR).....	38
2.4.5. Causa Comum.....	39
2.5. TIPOS DE REDUNDÂNCIA.....	40
2.5.1. Redundância de <i>Hardware</i> .....	41
2.5.2. Redundância de <i>Software</i> .....	43
2.5.3. O Impacto da Redundância.....	45
2.6. SISTEMAS EMBARCADOS.....	46
2.6.1. Sistema de Tempo Real.....	47
2.6.2. Windows CE.....	48
2.7. SISTEMA DE CONTROLE.....	51
2.7.1. Malha Aberta.....	52

2.7.2. Malha Fechada.....	53
2.7.3. Controlador PID.....	54
2.7.4. Controle em Tempo Real .....	55
<b>3. MATERIAIS E MÉTODOS.....</b>	<b>56</b>
3.1. INTRODUÇÃO.....	56
3.2. ESCOLHA DA TECNOLOGIA E DA ARQUITETURA .....	57
3.2.1. SIS Pneumáticos.....	58
3.2.2. SIS a Rele .....	59
3.2.3. SIS de Lógica Fixa em Estado Sólido .....	60
3.2.4. SIS Microprocessado (CLP).....	62
3.3. PROPOSTA DE UM SISTEMA TOLERANTE A FALHAS.....	64
3.3.1. Planta Didática .....	67
3.3.2. Comparação do Sistema com CLP Industrial de Controle .....	70
3.3.3. Obtenção de Dados para o Cálculo da PFD .....	71
3.4. AQUISIÇÃO DA PFD.....	71
<b>4. IMPLEMENTAÇÃO E RESULTADOS .....</b>	<b>74</b>
4.1. CANAL DE AQUISIÇÃO .....	74
4.2. CANAL DE VOTAÇÃO .....	77
4.3. CANAL DE PROCESSAMENTO .....	81
4.3.1. Criando a Imagem do Windows CE.....	81
4.3.2. Criando o Disco de <i>BOOT</i> .....	83
4.3.3. Efetuando o <i>BOOT</i> Local .....	83
4.3.4. Criando um Novo Projeto .....	83
4.4. SISTEMA DE CONTROLE .....	84
4.5. SISTEMA DE INTERTRAVAMENTO.....	90
<b>5. CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>96</b>
<b>REFERÊNCIAS.....</b>	<b>99</b>
<b>ANEXO A.....</b>	<b>103</b>
<b>APÊNDICE A .....</b>	<b>105</b>
<b>APÊNDICE B .....</b>	<b>107</b>

## CAPÍTULO 1

### 1. INTRODUÇÃO

O projeto de sistemas instrumentados de segurança ou sistemas de intertravamento de segurança (*shutdown*) era relativamente simples há trinta anos atrás (CCPS,1993), sem falar que não havia normas industriais sobre o assunto. De um modo geral a escolha recaía sobre reles. Mas a evolução não ocorre por acaso, nem apenas do desenvolvimento da consciência de que os acidentes podem e devem ser evitados. Infelizmente, é a partir de acidentes ocorridos nas indústrias de processos que se verificou que os sistemas de segurança normalmente empregados nas instalações industriais deixavam a desejar.

Após alguns acidentes fatais, como perdas de vidas humanas e agressão ao meio ambiente começaram-se a por em prática o conhecimento em segurança e o desenvolvimento de novos sistemas. Isso refletiu em normas nacionais e internacionais no desenvolvimento de produtos (*hardware* e *software*) específicos para serem usados na concepção e na arquitetura de novos projetos de instalações.

Os trabalhos nesta área começaram há cerca de dez anos, tanto nos Estados Unidos como em outros países. O tema comum nas normas recentes é: “quanto maior o risco do processo, tanto melhor devem ser os sistemas necessários para controlar tal risco” (GRUHN;CHEDDIE, 2006).

Hoje em dia, existe uma gama grande de escolhas de possíveis tecnologias (por exemplo, tecnologias distintas, níveis de redundância, etc.), que as pessoas ficam confusas. É comum que inicialmente projetistas adotem uma postura muito conservadora, e demonstrem interesse em instalar os sistemas para atingir um desempenho SIL 3. Infelizmente, quando eles percebem que os sistemas que são certificados para atender este nível são caros, eles acabam precisando encontrar uma outra saída, por exemplo, reavaliar a malha de controle para tentar baixar o SIL (Nível de Integridade de Segurança), o que gera um retrabalho e custos adicionais de projeto para as empresas.

Os sistemas SIL 1 proporcionam o menor nível de desempenho (de 90 a 99% de disponibilidade), basicamente para atender aos baixos níveis de risco de processos. Os sistemas SIL 3 oferecem os níveis de desempenho mais altos (de



99,9 a 99,99% de disponibilidade), para atender aos níveis mais elevados de risco do processo. As normas não obrigam e nem recomendam qual tecnologia deve ser usada, nem níveis de redundância ou intervalos de testes manuais devem ser adotados para atender os requisitos de desempenho para níveis de integridade distintos. Cabe ao projetista determinar quais sistemas foram projetados para atender às condições de segurança. Alguns livros de confiabilidade como Xai, Dai e Poh (2004) e Pham (2003), bem como, o relatório TR 84.01 da ISA e a norma IEC 61508 mostram métodos para o cálculo do desempenho de segurança de um sistema.

Os sistemas a reles têm condições de atender aos requisitos SIL 3, mas é raro encontrar, hoje em dia, alguém disposto a usar reles para grande sistemas. Os sistemas baseados em *software* e *hardware* com redundância dupla e tripla também são capazes de atender aos requisitos SIL 3. Esses sistemas também são certificados por órgãos independentes como TUV e Exida.

A sugestão deste trabalho é partir de um *hardware* e de um *software* existente e exaustivamente testado pelas diversas aplicações que o envolvem. A evolução dos PCs acarreta uma alta confiabilidade de *hardware* e o surgimento dos sistemas embarcado de tempo real vem aumentando as possibilidades de se ter novos sistemas desenvolvidos a partir da arquitetura do PC que apresentem uma alta disponibilidade e confiabilidade.

A questão é que, dentro de sistemas dedicados para SIL o processamento é feito com microcontroladores ou microprocessadores que estão disponíveis no mercado. Em sistemas mais antigos, pode-se encontrar microprocessadores inferiores aos existentes e utilizados hoje em dia nos PCs. Então, a indústria de sistema é obrigada a desenvolver todo um *hardware* específico e dedicado que tem um alto custo de desenvolvimento para atingir a disponibilidade exigida. E dentro destes *hardwares* existem *softwares* embarcados denominados *firmwares* que servem para rodar o programa aplicativo desenvolvido pelo usuário.

O objetivo é utilizar a arquitetura de um PC, cuja diferença reside no sistema operacional. Para tanto, utilizar-se-á um sistema operacional de tempo real, que pode ser comparado com o *firmware* dos sistemas dedicados. Será criado um aplicativo que irá rodar em cima deste *firmware*, podendo este aplicativo ser comparado com o *software* desenvolvido pelo usuário.

Para que seja possível a interface do aplicativo desenvolvido com os equipamentos de campo (sensores, elementos finais, etc.) serão desenvolvidas placas que se comunicarão com o PC via um protocolo serial.

Feito isto, as normas de certificação de sistemas de intertravamento de segurança serão seguidas para que seja possível o cálculo de disponibilidade do sistema proposto, visando à avaliação de todo este trabalho e validação do mesmo.

### 1.1. OBJETIVOS GERAIS

Estudar os diferentes sistemas de redundância existentes no mercado (sistemas pneumáticos, a reles, lógica fixa e controladores lógicos programáveis), verificar seus funcionamentos e benefícios, analisar as vantagens e desvantagens de tais sistemas, com isto, propor um sistema concorrente de baixo custo, mas tão eficaz quanto. Para tanto, implementar um sistema utilizando-se de um *hardware* padrão (PC) e um sistema *embedded* sem precisar desenvolver um sofisticado novo *hardware*.

### 1.2. OBJETIVOS ESPECÍFICOS

Os objetivos específicos são:

- Checar os vários sistemas disponíveis no mercado;
- Propor um sistema de baixo custo com confiabilidade;
- Comparar o sistema proposto com sistemas industriais;
- Aplicar o sistema proposto em um processo real;
- Calcular a disponibilidade do sistema proposto;
- Encontrar o SIL deste sistema de acordo com a IEC 61508.
- Verificar se o sistema proposto atende as exigências industriais, seguindo a prescrição da IEC 61508.

### 1.3. ESTRUTURA DA DISSERTAÇÃO

O restante desta dissertação está organizada em 4 capítulos. No Capítulo 2 faz-se uma revisão da literatura sobre os principais conceitos sobre nível de integridade de sistemas, sistema operacional de tempo real. No Capítulo 3, descrevem-se em detalhes o desenvolvimento do sistema redundante proposto. No Capítulo 4 relatam-se os resultados obtidos. E, finalmente, o Capítulo 5 apresenta a discussão dos resultados, as conclusões do trabalho e as propostas de trabalhos futuros.

### 1.4. ETAPAS METODOLÓGICAS

A metodologia desta pesquisa está apoiada na elaboração de um protótipo fundamentado pela IEC 61508 que é a norma que certifica este tipo de equipamento. A validação do mesmo foi baseada na norma em questão, para tanto, dois experimentos serão desenvolvidos, o primeiro focando uma aplicação em controle de processo e o segundo para um sistema de intertravamento. A Figura 1 ilustra essas etapas metodológicas.

As etapas metodológicas da pesquisa tem por objetivo as seguintes atividades e resultados:

1. Normas, SIL e redundância: revisão bibliográfica para encontrar a forma que é aplicada a IEC 61508 para encontrar o SIL de uma malha de controle.
2. Pneumático, Rele, Lógica fixa, CLP redundante: Analisar as tecnologias existentes no mercado afim de elencar as suas vantagens e desvantagens.
3. Canais de aquisição, votação e processamento: propor um sistema redundante a falhas com uma configuração 2oo3.
4. Validação em uma malha de controle: aplicação em um processo real a fim de comparar o sistema proposto com um sistema industrial.

5. Validação em um sistema de intertravamento: obter as taxas de falhas do sistema proposto para encontrar a disponibilidade, confiabilidade e o SIL deste sistema de acordo com a IEC 61508.

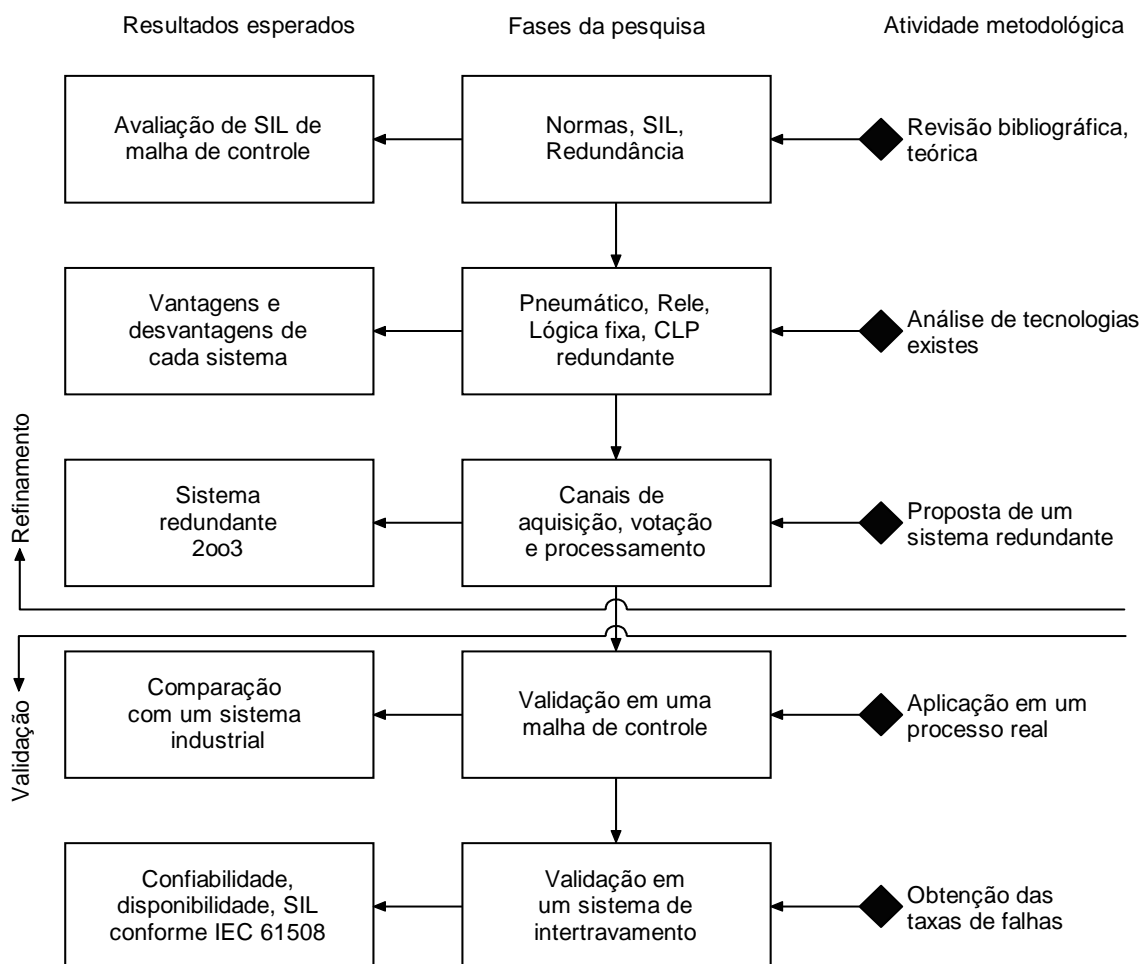


FIGURA 1 – ETAPAS METODOLÓGICAS.

## CAPÍTULO 2

### 2. REVISÃO DA LITERATURA

#### 2.1. OS SISTEMAS DE SEGURANÇA NOS PROCESSOS INDUSTRIAIS

Os sistemas de *shutdown* (ESD, SIS) para processos industriais têm evoluído bastante nestas últimas décadas, em termos de concepção, projeto e implementação. A evolução não é fruto do acaso, nem apenas do desenvolvimento de uma consciência de que os acidentes podem e devem ser evitados. Infelizmente, é a partir da análise de acidentes realmente ocorridos que se tomou conhecimento de que os sistemas de segurança normalmente empregados em instalações industriais deixavam a desejar (FINKEL *et al.*, 2006).

Um fator importante, para a ocorrência do acidente, é o excesso de autoconfiança, não só entre operadores, mas principalmente, em níveis gerenciais. Para não parar o processo, frequentemente assumem-se riscos desnecessários, e frequentemente o resultado é o acidente. Antes do ocorrido em Chernobil, um primeiro ministro soviético disse textualmente: “Nossos reatores nucleares são tão seguros que poderia-se instalar um aqui na Praça Vermelha, no centro de Moscou” (FINKEL *et al.*, 2006).

Após o acidente na plataforma de Piper Alfa ficou provado que não se pode confiar na capacidade de decisão/atuação do ser humano frente ao *stress* de um acidente grave. Posteriormente, estudos revelaram que, estatisticamente, quando submetidos à forte fator de *stress*, seres humanos tomam iniciativas erradas em 99% dos casos. Conclusão: o sistema de desligamento de emergência não pode depender de acionamento manual, devem ser acionados automaticamente (FINKEL *et al.*, 2006).

O excesso de confiança também atinge os sistemas de segurança, que muitas vezes não são exaustivamente testados com a frequência necessária para garantir uma probabilidade de falha suficientemente pequena. O problema surge na hora de testar os elementos finais. A única maneira de saber se uma válvula de

*shutdown* realmente está em boas condições de atuação, e que provavelmente vai fechar mesmo em caso de necessidade, é comandar a válvula para fechar e aguardar o seu fechamento completo. Mas infelizmente, isso é caro porque provoca a parada do processo (GRUHN; CHEDDIE, 2006).

## 2.2. NORMAS ATUAIS PARA SIS

Segundo Gruhn e Cheddie (2006), as normas desempenham vários papéis na execução de um projeto, principalmente em SIS. Como exemplo, pode-se citar o auxílio à equipe de projeto, no sentido de garantir que o produto em desenvolvimento obedeça a um determinado nível mínimo de qualidade e a função de seleção de métodos de projeto de eficiência reconhecida.

Outras funções que podem ser citadas são promover uniformidade entre diversas equipes de trabalho, prover guias de projeto, além de proporcionar uma base legal no caso de disputas judiciais.

Para se certificar um sistema, normalmente, faz-se necessária a utilização de normas apropriadas a cada aplicação. Algumas normas são genéricas e outras se aplicam a casos particulares. Nos próximos itens são descritos os principais objetivos de algumas das normas mais utilizadas no desenvolvimento de Sistemas de Segurança Instrumentado.

De acordo com Finkel *et al.* (2006), as normas sobre Sistemas Instrumentados de Segurança (SIS) têm como denominador comum não serem normas prescritivas e sim orientadas para exigir que se atinja um nível de desempenho desejado pelo sistema. Elas dizem o que precisa ser feito, mas não como fazê-lo. Exige do profissional que as usa um conhecimento de causa bem maior do que quem queira simplesmente seguir uma receita de "como se projeta" um sistema. A norma pode ser aplicada com qualquer tipo de tecnologia existente ou futura. Assim a tendência é não ser necessária se revisar a norma a cada vez que surgir uma nova tecnologia aplicável a sistemas de segurança, o que seria indesejável, levando-se em conta o tempo de estudo, maturação e aprovação destas normas.

A exceção é a norma N-2595 (Critérios de projeto e manutenção para sistemas instrumentados de segurança em unidades industriais) onde se tenta manter uma prescrição de como projetar um Sistema de Segurança, inclusive sugerindo a tecnologia a ser escolhida em função da classificação do risco envolvido em cada malha. Assim: para malhas classe I ou II pode-se implementar o *shutdown* no SDCD (Sistema Digital de Controle Distribuído), enquanto para malhas classe III e superiores, é preciso implementar a segurança em equipamentos independentes do SDCD.

Para malhas classe III, o sistema pode ser implementado em reles (só para sistemas simples ou pequenos) ou em CLP (Controlador Lógico Programável) de segurança desde que aprovado para esta classe pelo TUV (*Technische Überwachungs Verein* - ANEXO A) Organizações de Inspeção Técnica, conforme Anexo A.

Para malhas de classe superior a III, o sistema deve ser desenvolvido em CLP de segurança, aprovado para a classe da malha considerada, pelo TUV.

A *Functional Safety - Safety Related Systems* - Norma IEC 61508, da Comissão Eletrotécnica Internacional (Européia), de 1996, abrange todos os tipos de indústrias, incluindo medicina, transporte, nuclear, etc., e cobrindo várias tecnologias como reles, lógica fixa em estado sólido e sistemas programáveis. A norma IEC 61511 é derivada desta, porém destinada para as indústrias de processamento em geral.

*Application of Safety Instrumented Systems for the Process Industries* - ISA TR 84.01, da ISA (*The International Society for Measurement and Control*), de 1996. Norma aprovada pela ANSI, levou 11 anos em elaboração, inicialmente pretendia cobrir apenas a parte lógica do sistema em lógica programável, mas acabou abrangendo também os dispositivos de campo e outras tecnologias. É considerada pelos Americanos como uma Norma Internacional. Pode vir a conflitar (ou se tornar obsoleta ou ser substituída) pela IEC- 61511.

A Diretriz PES *Guidelines (Programmable Electronic Systems for Use in Safety Related Applications)* da Secretaria de Saúde e Segurança do Reino Unido, de 1997. Foi uma das primeiras normas publicadas a respeito do uso de PLCs em segurança, e embora focada em Sistemas Programáveis, é aplicável a outras tecnologias. Foi ponto de partida para a elaboração de outras normas européias e americanas.

*Guidelines for Safe Automation of Chemical Processes*, do Centro para Segurança do Processo (CCPS) do Instituto Americano de Engenheiros Químicos (AIChE) de 1993. Discorre sobre o uso de SDCDs (Sistema Digital de Controle Distribuído) e sistemas de intertravamento. Gerado por usuários, sem a interferência de fornecedores, o que permitiu uma velocidade de trabalho superior à usual neste tipo de comitê.

DIN/VDE 0801 é uma norma Alemã, apenas para os fabricantes dos sistemas. Detalha exigências para fabricação baseados nos riscos calculados conforme a norma DIN/VDE 19250, que por sua vez deve ser substituída pela IEC 61508. Os alemães têm uma agência de certificação independente, que é praticamente a única reconhecida mundialmente, para sistemas de segurança, a TUV.

Uma diferença significativa entre a IEC 61508 e a DIN/VDE 19250 é a classificação dos níveis de integridade de segurança, que pode ser observada na Tabela 1.

TABELA 1 – NÍVEL DE INTEGRIDADE DE SEGURANÇA IEC 61508 VERSUS DIN/VDE 19250

IEC 61508	Arquitetura	DIN/VDE 19250
SIL 4	Lógica fixa, falha segura	AK 8 AK 7
SIL 3	1oo2D, 2oo3	AK 6 AK 5
SIL 2	1oo1D	AK 4
SIL 1	1oo1 com WDT	AK 3 AK 2
Não definido	Não definido	AK 1

## 2.3. ANÁLISE DO RISCO

### 2.3.1. Perigo (*Hazard*)

O Instituto Americano de Engenheiros Químicos (AIChE) define perigo como uma característica inerente físico ou químico que tem o potencial para causar danos às pessoas, bens ou o ambiente. Ele é a combinação de um material perigoso, um



ambiente operacional, e certos acontecimentos não planejados que podem resultar em um acidente.

Os perigos estão sempre presentes. Por exemplo, a gasolina é um combustível líquido. Enquanto não existe uma fonte de ignição, a gasolina pode ser considerada relativamente benigna. Nosso objetivo é minimizar ou eliminar eventos ou acidentes perigosos. Por isso, não devemos armazenar gasolina perto de fontes de ignição (GRUHN;CHEDDIE, 2006).

A N-2595 diz que perigo é uma causa potencial de dano à integridade física e saúde, patrimônio, meio ambiente ou perda de produção (PETROBRAS, 2002).

### 2.3.2. Risco

De acordo com a N-2595 risco é a combinação da taxa de perigo com as consequências do evento perigoso (PETROBRAS, 2002).

O risco normalmente é definido como a combinação da gravidade e da probabilidade de um evento perigoso. O risco pode ser avaliado qualitativamente ou quantitativamente (GRUHN;CHEDDIE, 2006).

Segundo Finkel *et al.* (2006) o risco é uma combinação de dois fatores:

- a probabilidade da ocorrência do evento indesejável;
- a consequência da ocorrência.

Assim, um evento desfavorável que ocorra com frequência, porém de consequências mínimas, pode corresponder ao mesmo nível de risco que um evento raríssimo, porém, de consequências catastróficas.

A aceitabilidade de um determinado nível de risco é determinada pelos benefícios associados à aplicação crítica e, conseqüentemente, seus riscos, bem como pelos esforços necessários para que se consiga a redução desses riscos. Riscos com consequências catastróficas e que ocorram frequentemente não são toleráveis em nenhuma hipótese. Por outro lado, riscos com consequências não significativas, mesmo com ocorrência frequente, podem ser aceitáveis.

A norma IEC 61508 classifica o risco em três níveis (Comissão Eletrotécnica Internacional (IEC), 1997):

- Risco Intolerável: as consequências do risco são intoleráveis e sua ocorrência não pode ser justificada;
- Risco Inaceitável: as consequências do risco são inaceitáveis, embora possam ser suportadas sob certas condições;
- Risco Negligenciável: as consequências do risco são insignificantes e podem ser desprezadas.

### 2.3.3. Conceitos de Risco

Tanto para Finkel *et al.* (2006) como para Gruhn e Cheddie (2006), um conceito bem difundido em Sistemas Instrumentados de Segurança, mas errôneo, diz que todos os processos vão para o estado seguro quando os equipamentos são desligados. Vamos analisar um exemplo, uma caldeira quando desligada deve ser ventilada internamente para a diluição de gases combustíveis até se atingir um nível de segurança, antes que se possa partir novamente, ou que a eventual concentração de gases quentes junto a algum ponto mais quente dentro da caldeira possa levar a uma ignição espontânea destes gases. É comum que ventiladores de tiragem tenham uma alimentação de emergência para poderem ser acionadas, mesmo que uma caldeira pare por perder sua alimentação elétrica normal.

Planta desligada não significa planta segura. O conceito que todo processo reverte a uma condição segura quando se corta a alimentação elétrica geral da planta, pode ser falso em muitas instalações.

Uma unidade industrial que entra em parada de emergência por uma falha no sistema de parada de emergência, na verdade, pode estar em uma condição bem menos segura do que o desejado. É mais seguro garantir que o SIS não provocará muitas atuações desnecessárias.

No caso de SIS, um determinado risco é aceitável se atingir um nível baixo o suficiente, de forma que reduções maiores não sejam justificáveis dos pontos de vista técnico e econômico. Esse é o princípio ALARP (*As Low as Reasonably*

*Possible*), ou seja, o risco deve ser tão baixo quanto o razoavelmente praticável ou implementável (IEC, 1997). É importante considerar que um risco não é aceitável se ele puder ser facilmente reduzido. Portanto, mesmo um sistema com um nível de risco muito pequeno pode ser considerado como inaceitável se o seu nível de risco puder ser facilmente reduzido. Similarmente, um sistema que tenha um nível de risco significativo pode satisfazer aos requisitos se ele oferecer benefícios suficientes e reduções do risco forem consideradas impraticáveis. A Figura 2 representa o princípio ALARP.

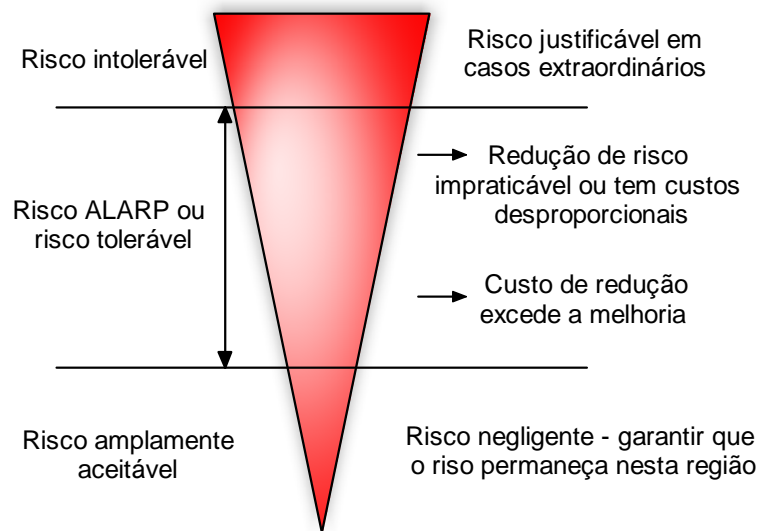


FIGURA 2 – ALARP.  
FONTE: INTECH BRASIL (2006).

O conceito do ALARP surgiu na Inglaterra, há algumas décadas, embora não tenha sido enunciado formalmente até a uns poucos anos. Entende-se que não há uma segurança total nem absoluta, o que se pretende fazer é sempre reduzir o nível de risco a um nível aceitável, e não a um hipotético risco igual a zero (FINKEL *et al.*, 2006).

Os riscos elevados não podem ser aceitos, há que reduzi-los, obviamente. Riscos muito pequenos podem até ser aceitos sem que se façam despesas adicionais para reduzi-los, pois não vale à pena realizar o investimento para reduzir riscos que em caso de acidentes, causem prejuízos pequenos. É mais barato correr o risco do que reduzi-lo. E riscos médios, nem tão pequenos que sejam desprezíveis, nem tão grandes que sejam absolutamente intoleráveis? Para estes a

técnica do ALARP recomenda simplesmente que se tome a decisão baseada no custo e benefício.

Pode-se exemplificar da seguinte forma, se por US\$ 3000 for possível reduzir um destes riscos a um nível aceitável, sempre valerá à pena fazer este investimento, mas se para esta função de segurança não muito crítica, for preciso gastar US\$ 1.000.000, talvez seja melhor economizar este dinheiro e simplesmente correr o risco, ou reduzi-lo a um nível ainda um pouco maior do que o que seria considerado bom. Imagine que ao classificar as funções de segurança quanto ao SIL, haja 20 funções em que se ficou em dúvida, SIL 0 ou SIL 1, enquanto outras 20 ficaram na balança entre SIL 1 e SIL2. Revisto o projeto acabaram todas 40 em SIL 1. SIL 1 nunca se refere a riscos de vida humana ou ferimentos sérios, etc., estes teriam sempre um SIL elevado. Se devido à condição particular daquela instalação cada malha SIL 1 custar, por exemplo, US\$ 50.000 devido a válvulas de grande diâmetro e material muito resistente à corrosão, etc (INTECH BRASIL, 2006).

Não faz sentido projetar o SIS para atender apenas às malhas que inicialmente se imaginava estarem entre SIL 1 e SIL 2, e economizar um milhão de dólares não instalando o SIS para as outras malhas? Obviamente, o risco de acidentes aumentou, mas não se está falando de acidentes fatais nem de ferimentos graves, perdas de equipamento caro de produção, etc., pois estes deveriam ter um SIL mais elevado (INTECH BRASIL, 2006).

#### 2.4. DEFINIÇÕES SOBRE SISTEMAS DE SEGURANÇA (SIS)

A norma ANSI/ISA TR-84.01 exige que a indústria determine um Nível de Integridade Desejado (SIL – *Safety Integrity Level*) para todas as aplicações de Sistemas de Instrumentados de Segurança (SIS – *Safety Instrumented Systems*). A determinação do SIL é uma decisão que passa pela análise de risco do processo, (PHA – *Process Hazard Analysis*), e inclui a avaliação do equilíbrio entre a probabilidade do evento, sua severidade e tolerância ao mesmo. Os métodos utilizados para a avaliação do SIL de uma malha controle, são:

- Lista de verificação (*Checklist*);
- E se (*What if*);
- E se / lista de verificação (*What if/checklist*);
- FMEA (análise do modo de falha e seus efeitos) (*failure mode and effects analysis*);
- FTA análise da árvore de falhas (*fault tree analysis*);
- HAZAN Analise de riscos (*Hazard Analysis*);
- HAZOP estudo de riscos e operacionalidade (*hazard and operability study*);
- FTA HAZOP.

Nos EUA, a OSHA (*Occupational Safety and Health Organization*) e a EPA (*Environment Program Agency*), através de seus programas PSM (*Process Safety Management*) e RPM (*Risc Manegement Program*) exigem que se faça uma análise de risco do processo, e a utilize como base para tomar medidas para a proteção dos trabalhadores, da comunidade e do meio ambiente. Um programa adequado deve incorporar a assim chamada “Boa prática de Engenharia”, que significa na prática seguir as normas e padrões emitidos por organizações tais como: ASME, API, ANSI, NFPA, ASTM, etc.

Em fevereiro de 1996, a ISA editou o Standard ISA TR-84.01 “Aplicação de Sistemas de Segurança Instrumentados nas Indústrias de Processo”. Em início de 1997, esta norma foi elevada à categoria de ANSI, ou seja, passou a ser de aplicação legalmente obrigatória. Obedecer esta norma, nos EUA não é uma opção, já que a não obediência pode resultar em ação civil e criminal, tanto contra as empresas como contra os indivíduos envolvidos (GRUHN; CHEDDIE, 2006).

Tanto a norma ANSI/ISA TR-84.01 como a IEC 61508 exigem que se defina um Nível de Integridade Desejado (SIL) para cada Sistema de Intertravamento de Segurança.

O SIL (Nível de Integridade de Segurança) é uma classificação de risco determinada a partir da probabilidade de falha sob a demanda. É muito importante levar em conta o impacto da confiabilidade e o modo de falha dos instrumentos de campo no SIL, o que frequentemente era esquecido quando se via um SIS apenas como o equipamento eletrônico contido no gabinete do sistema de *shutdown*. É

possível correlacionar um SIL às exigências de seus respectivos SIS. A IEC 61508 reconhece 4 níveis de SIL, enquanto a TR-84.01 de 1996 só reconhece os níveis de 1 a 3 (FINKEL *et al.*, 2006).

#### 2.4.1. Falhas

Se analisado com atenção a ISA TR-84.01, as falhas podem ser caracterizadas das seguintes formas:

- Falhas aleatórias: uma falha espontânea de componente (*hardware*). As falhas aleatórias podem ser permanentes (existem até serem eliminadas) ou intermitentes (ocorrem em determinadas circunstâncias e desaparecem em seguida).
- Falhas sistemáticas: uma falha escondida dentro do projeto ou montagem (*hardware* ou tipicamente *software*) ou falhas devido a erros (incluindo-se enganos e omissões) nas atividades de ciclo de atividades de segurança que fazem o SIS falhar em determinadas circunstâncias, sob determinadas combinações de entradas ou sob uma determinada condição ambiental.
- Falha em modo comum: o resultado de um defeito em modo comum.
- Defeito em modo comum: uma única causa que pode causar falhas em vários elementos do sistema. A única causa pode ser interna ou externa ao sistema.

As falhas detectáveis são aquelas que o próprio SIS consegue observar sua ocorrência, sinalizando tal fato e permitindo que se executem as devidas ações corretivas. Falhas não detectáveis são aquelas não percebidas pelos mecanismos de detecção e que permanecem residentes no SIS.

Outra classificação das falhas, que ocorrem em SIS, dada por Globe e Cheddie (2005), relaciona-se aos efeitos que tais falhas possam acarretar. Se as consequências forem inseguras, a falha é dita como sendo insegura. Por outro lado,

se as consequências não provocarem situações inseguras, a falha é classificada como sendo segura.

Um sistema implementado através de uma arquitetura tolerante a falhas, normalmente supõe a existência de módulos de controle e intertravamento redundantes. A ocorrência de uma falha do tipo inseguro e não detectável em um dos módulos da arquitetura permanece no sistema sem, no entanto, causar situações inseguras, pelo menos em um primeiro momento. Por outro lado, podem ocorrer falhas compensatórias em outros módulos, ou seja, falhas que afetem o funcionamento dos demais módulos do sistema da mesma forma que a falha insegura do primeiro módulo.

Desta forma, o dispositivo comparador da saída de cada módulo irá ser induzido a produzir saídas incorretas, gerando condições inseguras para o SIS. Se este não dispuser de módulos redundantes, a falha inicialmente descrita, por si só, já será capaz de produzir estados inseguros no SIS.

No entanto, há uma consideração que ameniza o impacto das falhas não detectáveis e inseguras no sistema. Os circuitos integrados com alta escala de integração possuem altas taxas de falhas, quando comparados com componentes convencionais e mesmo circuitos integrados com pequena e média escala de integração. No entanto, apenas uma pequena parcela dessa taxa de falhas refere-se a situações que irão provocar estados inseguros no sistema. Isto ocorre porque a maioria das falhas acaba por desabilitar completamente o funcionamento de um circuito integrado, e apenas uma parcela mínima dessas falhas irá, de fato, estabelecer condições inseguras no SIS.

A N-2595 define as falhas da seguinte forma (PETROBRAS, 2002):

- *Fail Safe* (Falha Segura): resultado de uma falha em um componente ou sistema, cujo estado final deve ser mais seguro ou menos perigoso.
- Falha: não cumprimento do serviço esperado de um dispositivo.
- Falha Espúria: falha cujo resultado implica ação de pelo menos um atuador, sem que tenha ocorrido realmente um evento iniciador que o demandasse.
- Falha na Demanda: falha que se caracteriza pela não ação ou ação incorreta de pelo menos um atuador, quando da ocorrência de um evento iniciador que demande essa ação.

- Falha Oculta: falha cuja ocorrência só é percebida quando a ação de uma malha de segurança é solicitada, seja por demanda ou teste.

#### 2.4.2. Análise Quantitativa e Qualitativa

Há duas formas principais para se efetuar uma Análise de Segurança, que são a análise do tipo qualitativo e a análise do tipo quantitativo. Antes que se possam obter valores numéricos, através de uma análise quantitativa, é indispensável à realização de uma análise qualitativa eficiente, sem a preocupação de se atribuir valores às ocorrências. Uma análise quantitativa não deve desviar o foco dos problemas existentes, tais como falhas de projeto. Uma das principais utilidades deste tipo de análise é se fazer uma comparação entre sistemas com funções similares.

A Análise Qualitativa procura identificar mecanismos que programem os requisitos especificados, de forma que se mantenha o nível de segurança de um Sistema Crítico durante sua operação (SEAMAN, 1999).

De acordo com Gruhn (2006) atualmente tem sido dada à preferência para a técnica HAZOP (*Hazard Operability*), sem dúvida a melhor para identificar riscos, porém demorada, cansativa e exige um grande volume de recursos. Se o processo que estiver sendo analisado for bem conhecido, talvez seja melhor usar a técnica de “listas de verificações” (*check list*). Se, por outro lado o processo envolver uma grande quantidade de equipamento mecânico interdependente pode ser melhor usar a técnica da Análise de Modos de Falhas e seus Efeitos (FMEA). Existe ainda a técnica que é preconizada pela API (*American Petroleum Institute*) chamada “Avaliação de Função de Segurança” (SAFE – *Safety Function Evaluation*), na qual se preenche um mapa Safe. É uma técnica semelhante à FMEA, já que aqui também se usa o modo de falha de cada equipamento.

A Análise de Risco deve também especificar a taxa aceitável de falha segura. A taxa de falha segura é normalmente chamada de “trip” falso, parada inconveniente ou taxa de “trips” espúreos. A taxa de falha segura ou espúrea é incluída na análise do sistema de segurança de um sistema, já que a partida e a parada de um processo são ocasiões com alta probabilidade de ocorrência de um evento perigoso.



É por isso que, em muitos casos, a diminuição de paradas espúreas aumenta a segurança do processo. A taxa aceitável de falha segura é comumente expressa como o tempo médio para uma falha segura ou espúrea ( $MTTF_{sp}$ ). As falhas seguras são também muito indesejáveis por reduzirem a produção, causar refugos e outros problemas que variam em função de cada processo considerado (FINKEL *et al.*, 2006).

#### 2.4.3. Probabilidade de Falha sobre Demanda (PFD)

Se o risco inerente ao processo for considerável aceitável, não existe necessidade de reduzi-lo, mas se ele for considerado maior do que aceitável é preciso reduzi-lo. Uma primeira atitude é procurar alterar o processo, ou seu controle, se isso não resolver, deve-se usar dispositivos de segurança autônomos, como válvula de alívio de pressão, discos de rupturas e etc.

Se tudo isso não resolver, então deve-se projetar o SIS para reduzir o risco a um nível aceitável. Este estudo de risco e redução do fator de risco (RRF – *Risk Reduction Factor*) antecede à especificação do SIS e deve ser feito para cada função de segurança (SIF - *Safety Instrumented Function*) (GRUHN;CHEDDIE, 2006).

PFD é a probabilidade de uma malha de segurança falhar em resposta a uma demanda (N-2595, PETROBRAS, 2002).

Segundo Finkel *et al.* (2006, p. 579), “O conceito de disponibilidade é bastante difundido. Assim, se um equipamento tem uma disponibilidade de 99%, significa que ele pode operar durante 99% do tempo disponível durante sua vida útil, e estará inoperante durante o 1% restante”.

Partindo da idéia de que poderá ocorrer um acidente quando houver uma demanda do SIS e ele estiver indisponível. Então, o risco de ocorrer um acidente fica dependendo da PFD e da frequência da demanda.

Então, pode-se dizer que:

$$PFD = 1 - D, \quad (1)$$

onde  $D$  é a disponibilidade.

$$RRF = \frac{1}{PFD}, \quad (2)$$

onde  $RRF$  é o fator de redução de risco.

De acordo com Gruhn e Cheddie (2006) SISs devem ser projetados para falhar em uma direção segura em caso de falha de um componente individual, perda de sinal, e perda da alimentação de energia (elétrica ou ar de instrumentação). Exceto para aquelas poucas aplicações necessitando de uma configuração de energizado para parar, exemplo: válvula de controle ar para fechar. SISs devem ser projetados para paralisar o processo em caso de falta de energia. Se a aplicação demanda circuitos que são energizados para parar, é necessário um diagnóstico especial e um sistema de suprimento de energia *back-up*.

A Tabela 2 mostra como são distribuídos os níveis do SIL.

TABELA 2 - SIL EM FUNÇÃO DE DISPONIBILIDADE E PROBABILIDADE DE FALHA SOB DEMANDA

Nível de integridade (SIL)	Disponibilidade (segura) desejada	$PFD$	$RRF$
4	99,99 %	0,01%	>10.000
3	99,90 a 99,99 %	0,01 a 0,1 %	1.000 a 10.000
2	99,00 a 99,90 %	0,1 a 1 %	100 a 1.000
1	90,00 a 99,00 %	1 a 10 %	10 a 100

FONTE: FINKEL *et al.* (2006) – IEC 61508.

A taxa de falha é normalmente representada pela letra grega minúscula lambda ( $\lambda$ ). É comum a utilização de unidades de "falhas por milhão de horas" ou "falhas por ano" (GLOBE;CHEDDIE, 2005).

Segundo Piazza (2000) a taxa de falhas é a frequência com que as falhas ocorrem num certo intervalo de tempo. É medida pelo número de falhas para cada hora de operação ou número de operações do sistema.

$$\lambda = \frac{\text{Número de falhas por ano}}{\text{horas por ano}}. \quad (3)$$

Goble (1998) demonstra que uma taxa constante de falha está relacionada com MTTF (tempo médio de falha) de acordo com a equação abaixo.

$$\lambda = \frac{1}{MTTF}. \quad (4)$$

*Mean Time to Repair (MTTR)* é um termo criado para incluir claramente tanto tempo de detecção de diagnóstico e tempo real de reparo. Na verdade *MTTR* é uma estimativa de tempo para, reconhecer e identificar a falha; tempo para obter peças sobressalentes; tempo para adquirir pessoas para a equipe; tempo real para fazer o reparo; tempo para documentar todas as atividades, e tempo para obter os equipamentos em funcionamento.

O tempo médio entre falhas (*MTBF*) é definido como a média de tempo de uma falha mais o tempo médio de reparação. Inclui o tempo de falha, o tempo necessário para detectar a falha, e tempo real de reparação.

$$MTBF = MTTF + MTTR. \quad (5)$$

$$\text{Disponibilidade} = \frac{\text{Tempo Disponível}}{\text{Tempo Total}}. \quad (6)$$

$$\text{Disponibilidade} = \frac{\text{Tempo Disponível}}{(\text{Tempo Disponível} + \text{Indisponível})}. \quad (7)$$

$$D = \frac{MTBF}{(MTBF + MDT)}, \quad (8)$$

onde *MDT* é o tempo médio indisponível.

$$\lambda_s = \frac{MTTF}{(MTBF + MTTR)}, \quad (9)$$

onde  $\lambda_s$  é a taxa de falha segura.

Para falhas perigosas, o tempo indisponível deve englobar não apenas o tempo para reparo, mas também o tempo de “descoberta”, ou seja, o tempo decorrido antes que se torne conhecimento da existência do problema.

$$\lambda_d = \frac{MTBF}{\left( MTBF + \left( \frac{TI}{2} \right) + MTTR \right)}, \quad (10)$$

onde  $TI$  é o intervalo de testes e  $\lambda_d$  é igual a falha perigosa.

Estas fórmulas somente são válidas para sistemas simples, não redundantes (Finkel *et al.*, 2006).

Para sistemas redundantes pode-se seguir a Tabela 3.

TABELA 3 – CÁLCULO DE  $PFD$  E  $MTTF_{sp}$  PARA SISTEMAS REDUNDANTES

Configuração	$MTTF_{sp}$	$PFD$
1 de 1	$1/\lambda_s$	$\lambda_d * (TI/2)$
1 de 2	$1/(2 * \lambda_s)$	$((\lambda_d)^2 * (TI)^2) / 3$
2 de 2	$1/(2 * \lambda_s^2 * MTTR)$	$\lambda_d * TI$
2 de 3	$1/(6 * \lambda_s^2 * MTTR)$	$(\lambda_d)^2 * (TI)^2$

FONTE: FINKEL *et al.* (2006) – IEC 61508.

#### 2.4.4. Análise de Sistemas

##### 2.4.4.1. Análise de um Sistema a Rele

Assumindo que um rele industrial tenha um  $MTBF$  de 100 anos. Assumir-se-á que se tenha um rele para cada entrada e saída no sistema. Irá se trabalhar com 8

entradas e 2 saídas, logo, irá se adicionar a taxa de falha segura de 10 reles. Assumindo um rele com 98% de taxa de falha segura.

$$MTTF_{sp} = \frac{1}{\lambda_s} = \frac{1}{\left(\frac{1}{100}\right) * 0,98 * 10} = 10,2 \text{ anos (10 anos)}.$$

Quando há uma demanda de desligamento do sistema, ela vem em uma entrada apenas, as oito entradas não são acionadas ao mesmo tempo. Logo, dever-se-á incluir apenas uma entrada e as duas saídas no modelo específico. Isso equivale a apenas três reles. Como os reles não têm diagnósticos, todas as falhas inseguras são inseguras e não detectáveis (GRUHN;CHEDDIE, 2006). A probabilidade de falha sobre demanda, o fator de redução de risco e a disponibilidade são calculados como mostrado abaixo:

$$PFD = \lambda_d * \left(\frac{TI}{2}\right)$$

$$PFD = \left(\frac{1}{100}\right) * 0,02 * 3 * \left(\frac{1 \text{ ano}}{2}\right)$$

$$PFD = 3e^{-4}$$

$$RRF = \frac{1}{PFD} = 3300$$

$$D = 1 - PFD$$

$$D = (1 - 3e^{-4})$$

$$D = 0,9997 = 99,97\%.$$

#### 2.4.4.2. Análise de um Sistema com Controlador Lógico Programável não redundante

Substituindo o sistema à rele por outro que utiliza um CLP de propósito geral, considerando:

*MTBF* da CPU = 10 anos.

*MTBF* do módulo de I/O = 50 anos.

Taxa de falha segura da CPU = 60%.

Taxa de falha segura do módulo de I/O = 75%.

$$MTTF_{sp} = \frac{1}{\lambda_s} = \frac{1}{\left(\left(\frac{1}{10}\right) * 0,6\right) + \left(\left(\frac{1}{50}\right) * 0,75 * 2\right)} = 11 \text{ anos,}$$

2: representa 2 módulos de I/O.

Assumindo um diagnóstico de cobertura de 90% da CPU e 50% para o módulo de I/O. Assumindo-se que o PLC é realmente testado manualmente anualmente.

$$PFD = \lambda_d * \left(\frac{TI}{2}\right)$$

$$PFD = \left[\left(\left(\frac{1}{10}\right) * 0,4 * 0,1\right) + \left(\left(\frac{1}{50}\right) * 2 * 0,25 * 0,5\right)\right] * \left(\frac{1 \text{ ano}}{2}\right)$$

$$PFD = 4,5e^{-3}$$

$$RRF = \frac{1}{PFD} = 220$$

$$D = 1 - PFD$$

$$D = (1 - 4,5e^{-3})$$

$$D = 0,9955 = 99,55\%.$$

Neste caso, verifica-se um desempenho de um sistema com CLP mais baixo do que um a rele. Assumindo que o CLP é realmente testado uma vez por ano (o que é um pressuposto excessivamente otimista para muitos sistemas). Um módulo de I/O com cobertura de diagnóstico de 50%, o que também é um pressuposto otimista para muitos sistemas de propósito geral. Mas, se percebe que CLPs de utilização geral são bons para intertravamento e ainda controle (GRUHN;CHEDDIE, 2006).

### 2.4.4.3. Análise de um Sistema de Redundância Tripla (TMR)

Assumindo os mesmos *MTBFs* de um CLP sem redundância e as mesmas taxas de falhas, pois o *hardware* é essencialmente o mesmo, só que no CLP redundante existe mais CPU e mais canais de comunicação. A configuração deste TMR é 2oo3.

$$MTTF_{sp} = \frac{1}{(6 * (\lambda_s)^2 * MTTR)}$$

$$MTTF_{sp} = \frac{1}{6 * \left[ \left( \left( \frac{1}{10} \right) * 0,6 \right) + \left( \left( \frac{1}{50} \right) * 0,75 * 2 \right) \right]^2 * \left( \frac{4 \text{ horas}}{8760 \text{ horas/ano}} \right)}$$

$$MTTF_{sp} = 45.000 \text{ anos.}$$

Adotando um diagnóstico de cobertura de 99% para a CPU e para o módulo de I/O, algo que poderá não acontecer em alguns sistemas. Assumindo que o CLP é realmente testado manualmente anualmente.

$$PFD = (\lambda_d)^2 * (TI)^2$$

$$PFD = \left[ \left( \left( \frac{1}{10} \right) * 0,4 * 0,01 \right) + \left( \left( \frac{1}{50} \right) * 2 * 0,25 * 0,01 \right) \right]^2 * \left( \frac{1 \text{ ano}}{2} \right)^2$$

$$PFD = 6,25e^{-8}$$

$$RRF = \frac{1}{6,25e^{-8}} = 16.000.000$$

$$D = 1 - (6,25e^{-8}) = 0,999999375 = 99,9999375\%.$$

Os cálculos indicam que o sistema TMR tem um tempo médio entre “trips” de 45.000 anos, e um fator de redução de risco superior a dez milhões, isto daria mais do que SIL 4. É importante advertir que estes não são certificados para uso em SIL 4. A norma IEC 61508 é cautelosa para nível de integridade maior que SIL 4 (GRUHN;CHEDDIE, 2006).

### 2.4.5. Causa Comum

Uma falha pode acarretar uma falha geral do sistema redundante. Exemplos ambientais típicos são calor, vibração, excesso de tensão, etc. Um método de quantificar causa comum é referido como o fator  $\beta$ . Este fator representa a percentagem de falhas identificadas em uma parte do sistema que pode impactar em canais múltiplos, e o sistema poderá falhar de uma só vez. Por exemplo, se um sistema redundante tem um fator  $\beta = 1\%$ , isso significa que, todas as falhas identificadas, das quais 1% poderia ocorrer em vários canais ao mesmo tempo e fazer todo o sistema falhar. 1% é suficiente para preocupar-se? E cerca de 10%? (FINKEL *et al.*, 2006).

Causa comum pode ser definida como um único problema que afeta múltiplos componentes. Como mostrado na Figura 3 (A, B e C), representam um sistema triplo. No entanto, se o item D falhar, todo o sistema falha. Este fator representa o percentual de falhas identificadas em uma fatia do sistema que irá afetar todo o sistema. O fator  $\beta$  foi derivado a partir de estudos empíricos. Smith (1997) sugere um intervalo típico para  $\beta = 20\%$  quando se utiliza componentes idênticos redundantes (como é feito com a maioria dos sistemas).

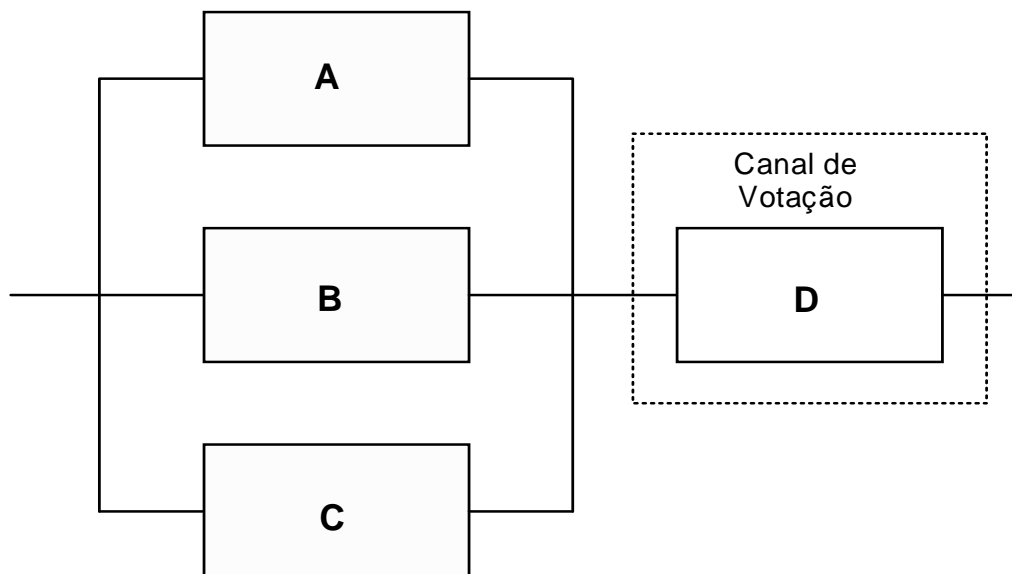


FIGURA 3 – CAUSA COMUM.  
 FONTE: GRUHN e CHEDDIE (2006).



Segundo Zio (2007), a confiabilidade de um sistema em paralelo é a probabilidade de um ou mais caminhos serem operacionais. Um sistema paralelo é um sistema em que não se considera haver falhas, a menos que todos os componentes tenham falhado. Como se observa na Figura 3, as falhas em um componente não influenciam na confiabilidade dos sistemas sobreviventes.

Uma configuração k-saída-de-n exige que pelo menos k módulos de um total de n devem estar operacionais para que o sistema esteja em funcionamento. Normalmente, um eleitor (votador) é necessário, se o eleitor é perfeito e todos os módulos possuem confiabilidade  $R$ , a equação para avaliar a confiabilidade desses blocos é (XIE; DAI; POH, 2004):

$$R = \sum_{i=k}^n \frac{n!}{i!(n-i)!} p^i (1-p)^{n-i}, \quad (11)$$

onde  $R$  é a confiabilidade do sistema,  $n$  é o número de unidades e  $p$  é a confiabilidade de cada unidade.

## 2.5. TIPOS DE REDUNDÂNCIA

Um componente é sujeito à falha em qualquer modo, aberto ou fechado. A redundância pode ser utilizada para aumentar a confiabilidade de um sistema sem qualquer alteração na confiabilidade dos componentes individuais que formam o sistema (PHAM, 2003).

Segundo Piazza (2000), a confiabilidade de um sistema é a probabilidade de que, quando em operação sob condições ambientais estabelecidas, o sistema apresentará uma performance desejada (sem falhas) para um intervalo de tempo especificado e a redundância é a existência de mais de um meio para se atingir um objetivo determinado.

Finkel *et al.* (2006) comenta que a redundância deve ser utilizada para fornecer um sistema em operação constante, embora um ou mais instrumentos possam falhar.

A N-2595 sugere que técnicas de utilização de diferentes tecnologias, projetos, fabricação, *software*, *firmware*, etc., podem ser usados para se reduzir à

influência das falhas de causa comum. Exemplos de métodos que podem ser utilizados para se obter a redundância diversa (PETROBRAS, 2002):

- medição de diferentes variáveis de processo, tais como pressão e temperatura, nos casos onde o relacionamento entre as variáveis é bem determinado e conhecido;
- uso de diferentes tecnologias de medição sobre a mesma variável de processo, tais como medição de vazão por vortex e coriolis;
- uso de diversidade geográfica, isto é, rotas alternativas para meios de comunicação redundantes.

Para Storey (1996), a forma mais utilizada para a prevenção dos efeitos de falhas é a utilização de módulos redundantes. A redundância de *hardware* implica inclusão de circuitos de *hardware* adicionais ao mínimo necessário para o funcionamento do sistema e a redundância de *software* implica geração de versões distintas do *software* do sistema ou de partes desse *software*, sempre se baseando em uma especificação comum.

#### 2.5.1. Redundância de *Hardware*

O emprego de redundância em um Sistema Instrumentado de Segurança consiste na utilização de componentes auxiliares com a finalidade de realizar as mesmas funções desempenhadas por outros elementos presentes no sistema. A finalidade principal da utilização de redundância nesses sistemas é a prevenção de condições ou estados inseguros.

Embora a redundância sempre implique na adição de novos componentes, deve-se fazer o possível para não aumentar a complexidade do sistema, de forma a não se ter efeito contrário, ou seja, diminuição da confiabilidade e da segurança do sistema. Conforme colocado em Johnson (1989), a redundância de *hardware* pode ser implementada através de três formas básicas:

- Redundância estática: utiliza o mascaramento de falhas como principal técnica. O projeto é feito de forma a não requerer ações específicas do sistema ou de sua operação em caso da ocorrência de falhas. Todos os elementos executam a mesma tarefa e o resultado é determinado por votação. Exemplos são TMR (*Triple Modular Redundancy*) e NMR (*N Modular Redundancy*);

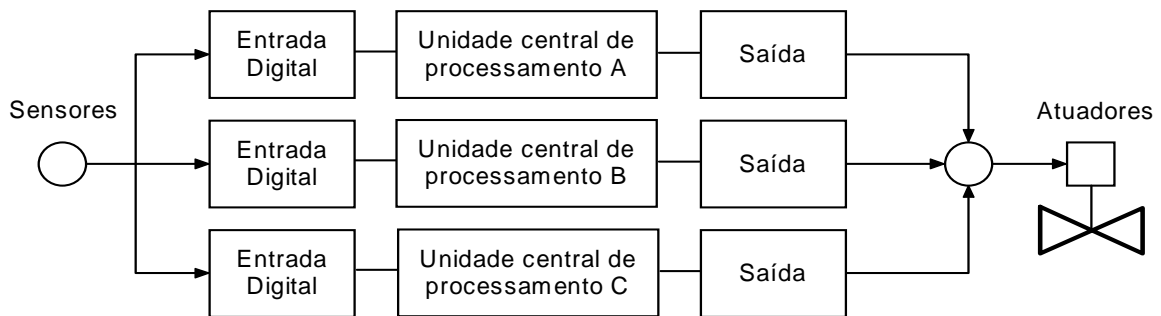


FIGURA 4 – SISTEMA DE REDUNDÂNCIA MODULAR TRIPLA.

Um sistema paralelo sério consiste de  $m$  caminhos paralelos e cada caminho possui  $n$  unidades conectadas em séries. Para calcular a confiabilidade utiliza-se a equação 12 (ZIO, 2007).

$$R = 1 - (1 - p^n)^m, \quad (12)$$

onde  $R$  é a confiabilidade e  $p$  é a confiabilidade de cada unidade.

- Redundância dinâmica: implica na detecção de falhas, caso em que o sistema deve tomar alguma ação para anular seus efeitos, o que normalmente envolve uma reconfiguração do sistema. Costuma ser usada em aplicações que suportam permanecer em um estado errôneo durante um curto período de tempo, tempo esse necessário para a detecção do erro e recuperação para um estado livre de falhas. Um exemplo de implementação de redundância dinâmica é através de módulos estepes (*hot standby*).

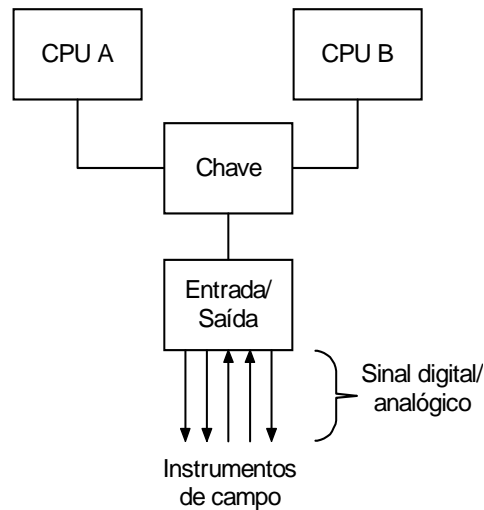


FIGURA 5 – SISTEMA *HOT STANDBY*.

- Redundância híbrida: consiste na combinação de técnicas estáticas com técnicas dinâmicas. Utiliza mascaramento de falhas para prevenir que erros se propaguem, detecção de falhas e reconfiguração para remover, do sistema, unidades com falha.

### 2.5.2. Redundância de *Software*

Simplesmente replicar *softwares* idênticos pode não ser uma estratégia muito eficaz. Rotinas idênticas de *software* vão apresentar erros idênticos. Logo, não basta copiar um programa e executá-lo em paralelo ou executar o mesmo programa duas vezes.

Um problema que surge é a falta de uma metodologia de eficiência reconhecida para a avaliação da segurança do *software* para os sistemas que precisam de segurança e que tem o *software* como componente crítico.

Pode-se dizer que a falta de experiência em especificações de software e das especificações em relação ao ambiente de aplicação representa um grave problema, fazendo com que o sistema atinja situações imprevistas, como consequência de procedimentos operacionais incorretos, de mudanças não esperadas no ambiente operacional, ou ainda de modos de falhas não previstas do sistema (JAFFE *et al.*, 1991).

Um caminho para se definir e avaliar melhor o conceito de segurança do *software* é através da definição de um conjunto de fatores que consigam representar adequadamente o conceito de segurança (KITCHENHAM; PELEEGER, 1996).

Em função da dificuldade da comprovação da não existência de falhas na implementação de um *software*, em relação à sua especificação, são utilizadas técnicas de redundância de *software*, cujo objetivo é tornar o *software* mais robusto em relação à segurança, ou seja, tolerante a falhas porventura ainda existentes (JOHNSON, 1989).

Segundo Burns e Wellings (1997) surge outras formas de redundância em *software*:

- **Diversidade:** também chamada programação diversitária, é uma técnica de redundância usada para obter tolerância à falhas em *software*. A partir de um problema a ser solucionado são implementadas diversas soluções alternativas, sendo a resposta do sistema determinada por votação (CHEN; AVIZIENIS, 1978).
- **Blocos de recuperação:** nessa técnica programas secundários só serão necessários na detecção de um erro no programa primário. Essa estratégia envolve um teste de aceitação. Programas são executados e testados um a um até que o primeiro passe no teste de aceitação. A estratégia de blocos de recuperação tolera  $n-1$  falhas, no caso de falhas independentes nas “ $n$ ” versões. Como mostrado na Figura 6:

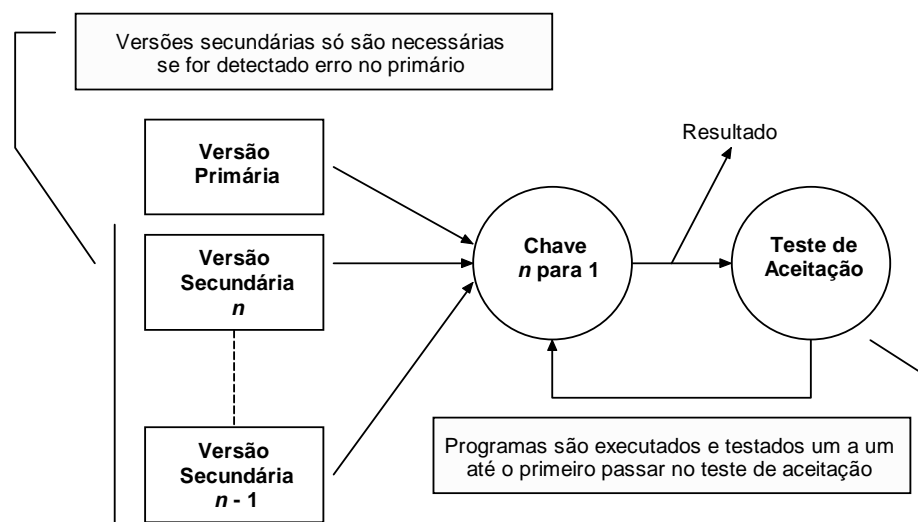


FIGURA 6 – BLOCOS DE RECUPERAÇÃO.

### 2.5.3. O Impacto da Redundância

De acordo com Gruhn e Cheddie (2006, p. 139), “Duplo nem sempre é melhor do que simples, e triplo nem sempre é melhor do que duplo.” Conforme mostrado na Figura 7,

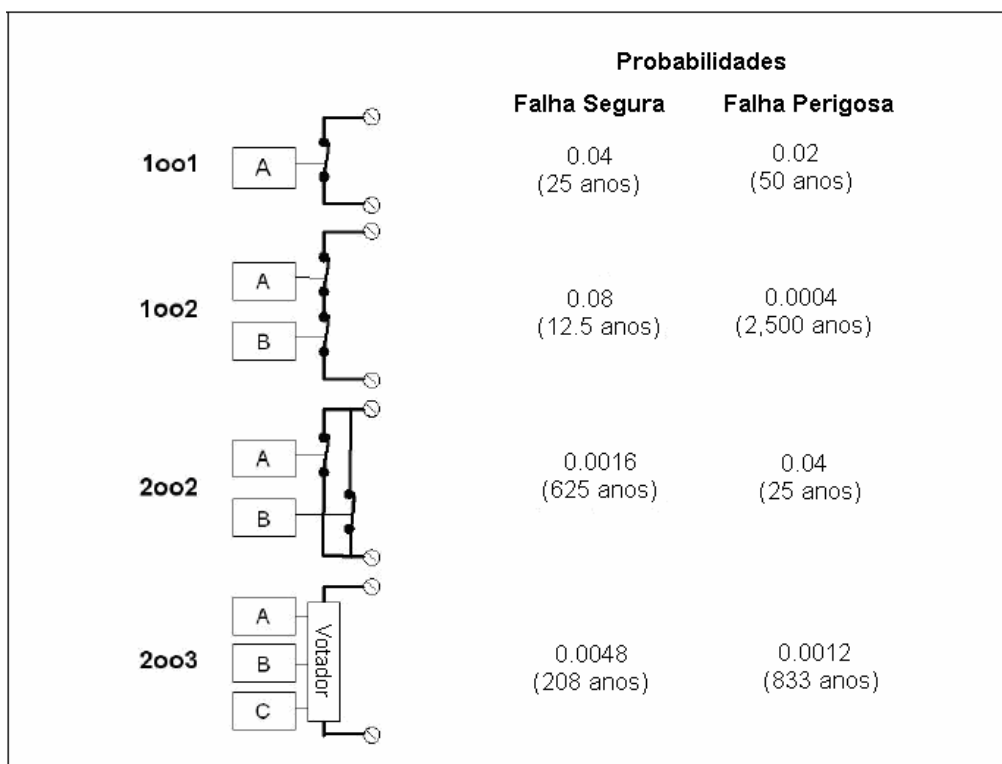


FIGURA 7 – O IMPACTO DA REDUNDÂNCIA.  
FONTE: GRHUN e CHEDDIE (2006).

Analisando um sistema sem redundância, 1001, conforme indicado na Figura 7, um exemplo de falha segura é quando o relé de contato abre e desenergiza o sistema causando um “trip”. Assumindo uma probabilidade de falha segura de 4% pelo período de um ano, significa que o sistema tem um *MTTF* (seguro) de 25 anos. Um exemplo de falha perigosa é quando o relé de contato não abre quando necessário não desenergizando o sistema. Assumindo uma probabilidade de falha perigosa de 2% pelo período de um ano, significa que o sistema tem um *MTTF* (perigoso) de 50 anos.

Um sistema 1002 tem suas saídas em série, isto quer dizer que o sistema precisa que pelo menos uma saída atue para que o sistema faça o *shutdown*, desta forma, a probabilidade de falha segura dobra para 8%, diminuindo o *MTTF* (seguro) para 12,5 anos. Mas, para ocorrer uma falha perigosa, ele precisa que 2 eventos

ocorram simultaneamente, então, a probabilidade de ocorrer uma falha perigosa é  $(0,02 * 0,02) = 0,0004$  o que faz com que o *MTTF* (perigoso) fique igual a 2.500 anos. Em outras palavras, um sistema 1oo2 é realmente muito seguro, mas aumenta a probabilidade de “trip” do processo.

Em sistemas 2oo2, os canais são ligados em paralelo, quando precisa-se desenergizar o sistema e um contato estiver colado, não irá ocorrer o *shutdown* caracterizando uma falha perigosa, como este sistema tem duas vezes mais hardware que um sistema simples, ele tem o dobro de probabilidade de ocorrer uma falha perigosa 4% que resulta em um *MTTF* (perigoso) de 25 anos. Para este sistema sofrer um “trip” é necessário que ocorra uma falha segura nos dois canais. A probabilidade de ocorrer dois eventos ao mesmo tempo é calculada pela probabilidade de um único evento ao quadrado, logo,  $(0,04 * 0,04) = 0,0016$  que corresponde a um *MTTF* (seguro) de 625 anos. Esta configuração reduz as paradas desnecessárias, mas aumenta o risco das falhas perigosas.

O Sistema 2oo3 é um sistema de votação por maioria, o que dois ou mais canais decidir é o irá acontecer com o sistema. Um sistema 1oo2 precisa ter duas falhas simultâneas para ocorrer uma falha perigosa, o 2oo3 também, como este sistema é triplo então ele tem três vezes mais dupla combinações de falha (A + B, A + C, C + B), logo, a probabilidade de falha perigosa dele é de  $3 * 0,0004 = 0,0012$  fornecendo um *MTTF* (perigoso) de 833 anos. Agora, em sistema 2oo2 para acontecer um “trip” ele necessita que aconteçam falhas seguras nos dois canais, novamente o sistema 2oo3 triplicou a dupla falha segura do sistema 2oo2 gerando uma probabilidade de  $3 * 0,0016 = 0,0048$  apresentando um *MTTF* (seguro) de 208 anos (GRUHN; CHEDDIE, 2006).

## 2.6. SISTEMAS EMBARCADOS

Sistemas embarcados são geralmente projetados para aplicações específicas, são o oposto dos sistemas construídos para aplicações genéricas. Os computadores pessoais são exemplos de sistemas de uso geral, projetados para atender várias aplicações distintas. São sistemas sujeitos a uma atualização contínua do *software* a ser executado, o que não ocorre nos sistemas embarcados,

uma vez programada uma aplicação específica, esta não sofrerá modificações ao longo da sua vida. Por exemplo, não se espera o melhoramento do *software* de um forno microondas após a sua venda. Contudo, com a evolução da complexidade dos sistemas embarcados, observa-se, cada vez mais, a necessidade de atualizações frequentes dos aplicativos a serem executados nos mesmos. Como o caso dos telefones celulares, que incorporam gradualmente diversas modificações no *software* executado.

A demanda por equipamentos inteligentes e soluções dedicadas, capazes de apresentar resultados eficientes para problemas, transforma a utilização de microprocessadores e sistemas embarcados em uma parcela importante do mercado de computação. Desta maneira, a demanda por sistemas operacionais embarcados que tenham a capacidade de comandar novos dispositivos e equipamentos é crescente e irreversível (ORTIZ, 2001).

Um sistema embarcado é uma junção do *hardware* e do *software* de um computador, projetados para executar uma tarefa específica (BARR, 1999).

### 2.6.1. Sistema de Tempo Real

É importante diferenciar Sistema de Tempo Real e Sistema Operacional de Tempo Real. Sistema de Tempo Real é um conjunto de todos os elementos especificados para essa finalidade, como *hardware*, sistema operacional e aplicativos. O Sistema Operacional de Tempo Real é um elemento do sistema completo de tempo real (OLIVEIRA, 2001).

Um sistema é classificado como de tempo real quando a execução de toda e qualquer tarefa devem se dar dentro de uma faixa de tempo estipulada *a priori*. Isto é importante para que o sistema possa realizar tarefas periódicas ou reagir a estímulos do meio sempre dentro de um tempo previsível. No caso da automação industrial esse tempo depende das constantes de tempo do processo a serem controladas (OLIVEIRA, 2001).

Aplicações com tempo real com restrições de tempo real são menos interessadas em uma distribuição uniforme dos recursos e mais interessadas em atender requisitos, tais como, períodos de ativação e *deadlines*. Essas aplicações



são usualmente organizadas na forma de várias *threads* ou tarefas concorrentes, logo um requisito básico para os sistemas operacionais de tempo real é oferecer suporte para tarefas e *threads* (OLIVEIRA, 2001).

As tarefas são abstrações que incluem: um espaço de endereçamento próprio (possivelmente compartilhado), um conjunto de arquivos abertos, um conjunto de direitos de acesso, um contexto de execução formado pelos registradores do processador.

As *Threads* são tarefas leves, únicos atributos são aqueles associados com o contexto de execução. Portanto, o chaveamento entre duas *threads* de uma mesma tarefa é muito mais rápido.

Uma aplicação tempo real é tipicamente um programa concorrente formado por tarefas e/ou *threads* que se comunicam e seguem um certo sincronismo. Existem duas grandes classes de soluções para programação concorrente: Troca de Mensagens e Variáveis Compartilhadas.

Os Sistemas de tempo real lidam com periféricos especiais, ou seja, diferentes tipos de sensores e atuadores, usados na automação industrial e controle de equipamentos em laboratório. O projetista da aplicação deve ser capaz de desenvolver os seus próprios *drivers* de dispositivos e incorporá-los ao sistema operacional (OLIVEIRA, 2001).

### 2.6.2. Windows CE

Windows CE (*Compact Edition*) é uma versão da popular linha de sistemas operacionais da Windows para dispositivos portáteis. É suportado no Intel x86 e compatíveis como, MIPS, ARM, e processadores Super Hitachi.

Algumas características do Windows CE (MICROSOFT, 2009):

- Compacto;
- Alta velocidade em relação a outros Sistemas Operacionais;
- Interface gráfica;
- Baixo custo da licença;

- Ocupa pouco espaço de memória;
- Portável de forma a rodar em diversos processadores e tipos de *hardware*;
- Modular (permitir uma adaptação rápida e fácil a um sistema particular);
- Compatibilidade com a API Win32;
- Disponibiliza processamento em tempo real (crítico em determinados sistemas embarcados);
- Implementa uma política agressiva de gestão de energia.

Oferece um controle industrial em tempo real com um trajeto de migração de custo flexível e baixo, controlando inclusive até o chão de fábrica.

Em primeiro lugar, é baseado no mesmo Win-32 API, o que significa que os usuários podem desenvolver aplicações usando exatamente as mesmas ferramentas. O Windows CE continua a tradição Microsoft de fazer o sistema virtualmente transparente. Isto significa que usuários e desenvolvedores de aplicações gastam mais tempo sobre a funcionalidade do que para aprender sobre o sistema operacional.

Em segundo lugar, o Windows CE tem uma tecnologia implementada chamada COM, que possibilita a comunicação com outros dispositivos.

A Microsoft considera que seu sistema operacional é um sistema de tempo real “*hard*” baseada na definição estabelecida pelo OMAC (*Open, Modular, Architecture Control*): “um sistema de tempo real *hard* é um sistema que falhará se seus requisitos de tempo não forem atendidos”. Estes sistemas são requeridos, sem falhas, para satisfazer todos os requerimentos de resposta no tempo a todo o momento sob qualquer circunstância e que se não realizados dentro de um tempo final de execução tem efeitos catastróficos ao processo.

“Um sistema de tempo real “*soft*” pode tolerar variações significantes no tempo de atendimento aos serviços do sistema como as interrupções, timers e o escalonador” (MICROSOFT, 2009).

Tacke e Ricci (2002) mensuraram a latência e o *jitter* utilizando esse sistema. A latência é uma medida do atraso que o sistema operacional leva para atender uma tarefa. O *jitter* mede a variação deste atraso. Esta análise permitiu tirar algumas

conclusões sobre que tipos de aplicação de tempo real podem ser implementadas utilizando o Windows CE. O teste realizado por Tacke e Ricci (2002) é bastante simples, é gerado um sinal que por sua vez gera uma interrupção, a rotina que trata esta interrupção atribui a um LED do circuito o valor '1', então gera um evento do Windows e em seguida coloca o mesmo LED em '0'. Também foi implementada uma aplicação, com a prioridade mais alta do sistema, que trata o evento do Windows que foi gerado pela rotina de tratamento da interrupção. Esta aplicação, da mesma forma, atribui o valor '1' a um LED e, em seguida, o coloca em '0'. A partir daí, foram feitas várias medições, utilizando um osciloscópio, para calcular a latência da rotina de interrupção e a latência da aplicação de teste. Foram feitas duas análises: uma onde a aplicação era executada sozinha; outra onde a aplicação dividia o processador com outra tarefa de mesma prioridade.

Este estudo permite concluir que, em aplicações onde às tarefas operam com intervalos de tempo da ordem de mili segundos ou até um pouco menores, o Windows CE apresenta um excelente tempo de resposta. Porém, se a aplicação apresentar tempos da ordem de micro segundos, um estudo mais detalhado deverá ser realizado sobre a arquitetura de *hardware* que será utilizada.

O Windows CE provê algumas tecnologias que são fundamentais para o desenvolvimento de aplicações de tempo real. Algumas delas são as seguintes (MICROSOFT, 2009):

- 256 níveis de prioridades para as "*threads*": é a partir desta flexibilidade disponibilizada pelo sistema que são implementadas as políticas de escalonamento.
- Interrupções aninhadas: permite que interrupções de prioridade mais alta sejam atendidas imediatamente ao invés de esperar que uma rotina de interrupção de prioridade mais baixa termine.
- *Per-thread quantum*s: permite que uma aplicação defina o quantum das threads. Inversão de prioridade. É uma situação onde o uso de um *mutex*, uma seção crítica, ou um semáforo por uma *thread* de menor prioridade impede a execução de uma *thread* de mais alta prioridade quando estas estão utilizando os mesmos recursos. Para corrigir este tipo de situação e garantir o funcionamento correto do sistema, o Windows CE permite que a *thread* de prioridade mais baixa, herde a

prioridade da *thread* de maior prioridade e utiliza-se a CPU com esta prioridade mais alta até que se termine de utilizar o recurso.

- MMU (*Memory Management Unit*): é um bloco de *hardware* que transforma endereços virtuais em endereços físicos. Na MMU, o valor no registro de realocação é adicionado a todo endereço lógico gerado por um processo na altura de ser enviado para a memória. O programa manipula endereços lógicos; ele nunca vê endereços físicos reais. A plataforma onde será colocada a imagem gerada pelo *Platform Builder* necessita da implementação antecipada ou já existente de MMU.

## 2.7. SISTEMA DE CONTROLE

Os processos industriais exigem controle na fabricação de seus produtos. Os processos são muito variados e abrangem muitos tipos de produtos, como por exemplo: a fabricação dos derivados do petróleo, produtos alimentícios, à indústria de papel e celulose, etc.

Em todos estes processos é absolutamente necessário controlar e manter constantes algumas variáveis, tais como pressão, vazão, temperatura, nível, PH, condutividade, velocidade, umidade, etc. Os instrumentos de medição e controle permitem manter constantes as variáveis do processo com os seguintes objetivos: melhoria na qualidade do produto, aumento em quantidade do produto, segurança e melhoria do meio ambiente.

Um sistema de controle consiste em subsistemas e processos (ou plantas) reunidos com o propósito de controlar as saídas do processo. Eles podem ser em malha aberta ou malha fechada. Na sua forma mais simples, um sistema de controle fornece uma saída ou resposta para uma dada entrada ou estímulo, conforme mostrado na Figura 8.



FIGURA 8 – SISTEMA DE CONTROLE.  
FONTE: NISE (2002).

Sistemas, que realizam medição e correção, são chamados de sistemas a malha fechada e sistemas que não têm essas propriedades são chamados de sistemas a malha aberta.

### 2.7.1. Malha Aberta

Um Sistema em malha aberta pode ser visualizado na Figura 9. Consiste em um subsistema chamado de transdutor de entrada, que converte a forma de entrada na usada pelo controlador. O controlador age sobre um processo ou planta. A entrada às vezes é chamada de referência ou *set-point* (*SP*), da mesma forma que a saída pode ser chamada de variável controlada ou manipulada (*MV*).

A característica que distingue um sistema a malha aberta é que este não pode compensar a ação de uma perturbação que sejam adicionadas ao sinal atuante do controlador.

Os sistemas a malha aberta, portanto, não corrigem os efeitos de perturbações e são comandados unicamente com base na entrada (NISE, 2002).

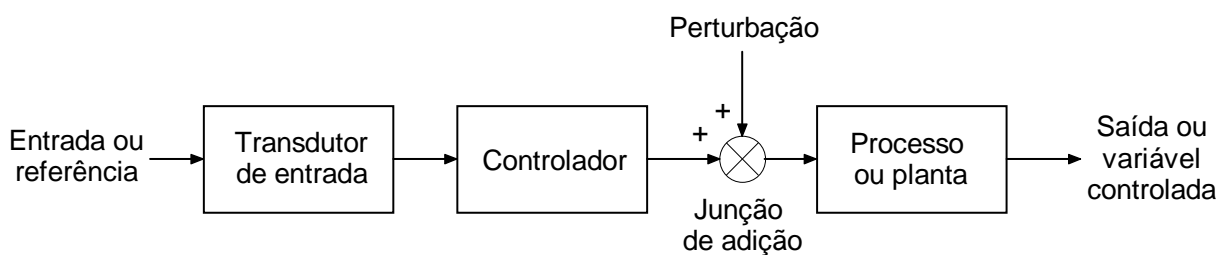


FIGURA 9 – SISTEMA A MALHA ABERTA.  
FONTE: NISE (2002).

### 2.7.2. Malha Fechada

Uma das desvantagens dos sistemas a malha aberta é a incapacidade de corrigir os efeitos das perturbações, que podem ser superadas nos sistemas a malha fechada. A arquitetura é mostrada na Figura 10.

O transdutor de entrada converte a forma da entrada na forma usada pelo controlador (*set-point SP*), um transdutor de saída, ou sensor, mede a resposta da saída e a converte na forma usada pelo controlador (variável do processo *PV*). A primeira junção de adição adiciona algebricamente o sinal de entrada ao sinal da saída, que chega pelo canal de realimentação, encontrando assim o sinal de erro.

O sistema a malha fechada compensa perturbações medindo a resposta de saída, retornando esta medição através de um sinal de realimentação e comparando essa resposta com a entrada da junção de adição. Se existir alguma diferença (sinal de erro), o controlador age sobre a planta, por meio de um sinal atuante (sinal de controle *CO*), para fazer a correção. Se não existir nenhuma diferença o controlador não irá atuar sobre a planta, uma vez que esta é a resposta desejada.

Os sistemas a malha fechada apresentam vantagens óbvias de uma maior precisão que os sistemas a malha aberta. Eles são menos sensíveis a ruídos, a perturbações e a mudanças nas condições ambientais. A resposta transitória e o erro de estado estacionário podem ser controlados de modo mais conveniente e com maior flexibilidade nos sistemas a malha fechada (NISE, 2002).

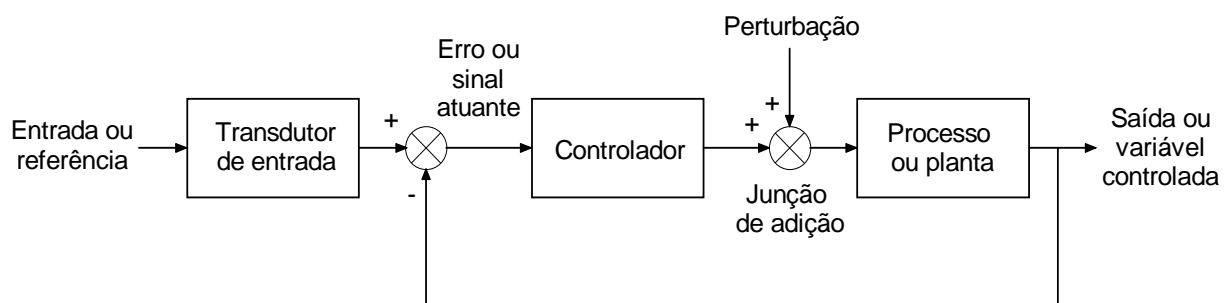


FIGURA 10 – SISTEMA A MALHA FECHADA.  
FONTE: NISE (2002).

### 2.7.3. Controlador PID

A combinação das ações Proporcional, Integral e Derivativa dá origem ao que denomina-se de controlador PID. O objetivo é aproveitar as características particulares de cada uma destas ações a fim de se obter uma melhora significativa do comportamento transitório e em regime permanente do sistema controlado. O sinal de controle gerado pelo controlador PID é assim genericamente dado em Nise (2002) como:

$$u(t) = K_p e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{d}{dt} e(t), \quad (13)$$

onde

$u(t)$  – Sinal de controle na saída do controlador (CO), no domínio do tempo;

$e(t)$  – Sinal de erro na entrada do controlador (erro), no domínio do tempo;

$K_p$  – Ganho Proporcional;

$T_i$  – Tempo Integral;

$T_d$  – Tempo Derivativo.

A função de transferência do controlador PID em Nise (2002) é dada por:

$$U(S) = K_p E(S) + \frac{K_i}{S} E(S) + K_d S E(S), \quad (14)$$

onde

$U(S)$  – Sinal de controle na saída do controlador (CO), no domínio da frequência;

$E(S)$  – Sinal de erro na entrada do controlador (erro), no domínio da frequência;

$K_p$  – Ganho proporcional;

$K_i$  – Ganho integral;

$K_d$  – Ganho derivativo.

A escolha das variáveis de controle ( $K_p$ ,  $K_i$ ,  $K_d$ ) depende do desempenho final do sistema. A escolha destes parâmetros é chamada de sintonia do controlador. Existem alguns métodos de sintonia que dependem do conhecimento do processo e

outros métodos que são matemáticos que levam em consideração a resposta do sistema a um sinal degrau, por exemplo, os métodos de sintonia de *Ziegler e Nichols* (NISE, 2002).

#### 2.7.4. Controle em Tempo Real

Segundo Loures (1999), controle em tempo real é a habilidade de necessariamente e sem falhas responder a um evento dentro de um período de tempo garantido. Por exemplo, na execução da malha de controle ilustrada na Figura 11, deve-se garantir que a saída da controlados responda ao valor de entrada medido dentro de um intervalo de tempo específico, conhecido como tempo de ciclo de malha de controle.

Se este tempo é constante, então o sistema de controle é estável e com comportamento determinístico; se varia não existe garantia quanto à estabilidade do sistema. Portanto, fica implícito no nome “sistema em tempo real” a necessidade de controle de tempo fornecido para que sejam lidos os sensores e atualizadas as saídas do sistema.

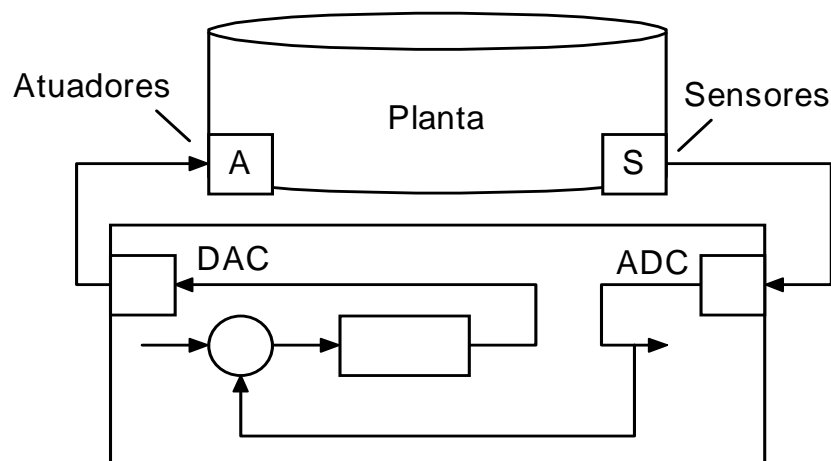


FIGURA 11 – SISTEMA DE CONTROLE CLÁSSICO.



## CAPÍTULO 3

### 3. MATERIAIS E MÉTODOS

#### SISTEMA TOLERANTE A FALHAS UTILIZANDO WINDOWS CE, FOCANDO CONTROLADORES INDUSTRIAIS.

##### 3.1. INTRODUÇÃO

No dia a dia da indústria de processo se faz mais que necessária à utilização de sistemas tolerantes a falhas, devido ao perigo e a complexidade que alguns processos apresentam. Logo, tem-se como objetivo, neste capítulo, apresentar uma proposta de um sistema tolerante a falhas baseado em um sistema operacional de tempo real, Windows CE.

Descreve-se então, a proposta do sistema de redundância tripla para a utilização em sistemas de segurança, *shutdown*, demonstrando os passos necessários para desenvolver um sistema com um baixo custo de hardware, uma vez que este sistema será baseado na arquitetura de um computador padrão, e ainda será demonstrada uma metodologia de escolha de SIS.

Após o desenvolvimento do mesmo irá ser calculada a PFD deste sistema, seguindo a IEC 61508, para que este seja validado nos mesmos moldes que os equipamentos industriais existentes hoje no mercado. Este, também será aplicado em um controle de processo, para isso, utilizar-se-á uma planta didática que ilustra um processo industrial, uma planta de nível.

### 3.2. ESCOLHA DA TECNOLOGIA E DA ARQUITETURA

Em projetos de sistemas de intertravamento de segurança deve-se inicialmente fazer a análise de risco da malha de controle, de modo a identificar o nível de integridade de segurança que aquela malha necessita. Tudo começa com a análise da malha e a definição do SIL. Feito este levantamento surgem algumas dúvidas em relação a qual equipamento deve-se utilizar.

Rapidamente poderia ser pensado em utilizar um sistema com entradas analógicas triplicadas, painel de lógica com redundância, no mínimo tripla, válvulas inteligentes de duplo bloqueio e ainda sistema de desligamento testado mensalmente. Mas, executar esta idéia é um tanto quanto difícil de ser implementada além de se ter um custo relativamente elevado. Ou ainda, fazer como os franceses fizeram há 200 anos, eles criaram uma lei que obrigava os fabricantes de dinamite morar com a sua família dentro da empresa, mas isto também não é possível.

Desta forma, devem-se fazer algumas considerações em relação à escolha dos equipamentos e da tecnologia a ser utilizada:

- Número de paradas indesejadas;
- Desempenho de segurança;
- Tamanho físico do equipamento e da instalação;
- Necessidade de testes;
- Custo do equipamento;
- Tecnologia do sistema:
  - Pneumático;
  - Reles;
  - Estado sólido;
  - Microprocessado.
- Tipo do sistema redundante:
  - Simples;
  - Duplo, 1oo2 ou 2oo2;
  - Triplo, 2oo3;
  - Com canal de diagnóstico.

Cada tecnologia tem vantagens e desvantagens. Não existe um sistema que seja o melhor. Não é tanto uma questão de quem é melhor, mas sim o que é mais adequado, com base em fatores como o orçamento, tamanho, nível de risco, a complexidade, flexibilidade, manutenção, interface de comunicação, requisitos de segurança, etc.

### 3.2.1. SIS Pneumáticos

Sistemas pneumáticos ainda estão em uso e ainda são perfeitamente adequados para determinadas aplicações. Uma aplicação muito comum para sistemas pneumáticos é a indústria *offshore*, onde os sistemas devem funcionar sem eletricidade. Sistemas pneumáticos são relativamente simples (assumindo que eles são pequenos). Eles, normalmente, são utilizados em pequenas aplicações onde há um desejo de simplicidade e a necessidade de segurança intrínseca.

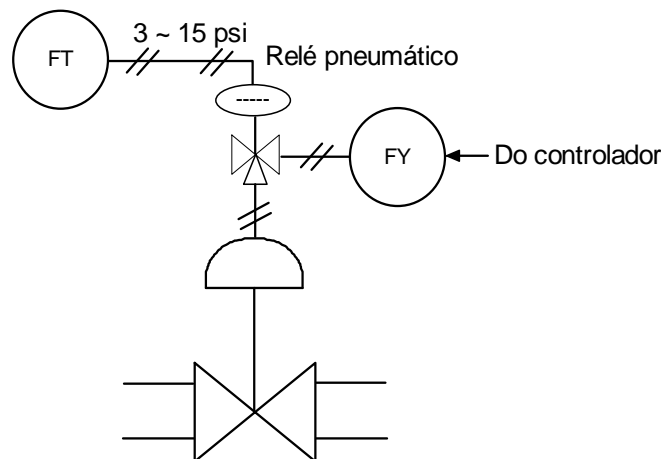


FIGURA 12 – SISTEMA DE PARADA PNEUMÁTICO.

Quando a saída do transmissor (FT) atingir um valor de parada, o rele pneumático alivia o ar do atuador, fazendo com que a válvula siga para uma condição de segurança, fechando-a.

Vantagens:

- Não precisa de eletricidade;
- Falha segura;
- Sistema confiável;

Desvantagens:

- Precisa de ar limpo e seco;
- Necessita ser testado frequentemente (mensalmente).

### 3.2.2. SIS a Rele

Reles especiais, chamados de “Rele de Segurança”, são construídos com múltiplas bobinas, proteção contra contato selado. Estes componentes normalmente trabalham energizados, ou seja, precisam ser desenergizados para desligar. São adotados quando a lógica de intertravamento é simples. Também é possível implementar o tipo energizado para desligar (trip) nas aplicações onde é fundamental evitar paradas indesejáveis.

Vantagens:

- Falha segura;
- Baixo custo inicial;
- Distribuído na planta;
- Imune a interferência;
- Várias tensões de alimentação;
- Velocidade de atuação rápida.

Desvantagens:

- Trips espúrios;
- Sem diagnósticos;
- Sem comunicação serial;
- Complexos para sistemas grandes;
- Reprogramação trabalhosa;
- Alto custo de vida.

A Figura 13 representa uma aplicação típica de utilização dos sistemas a reles, cada rele é ligado a uma chave de segurança (vazão – FS, nível – LS e pressão – PS), onde seus contatos são ligados em série com uma botoeira de *shutdown* de emergência. Uma fonte de alimentação é ligada neste circuito para alimentar a válvula solenóide que permanece energizada para ficar na posição de aberta. Se a botoeira de emergência for pressionada ou se faltar energia, a válvula solenóide irá para a posição de fechada. Ainda temos uma lâmpada que indica quando o sistema está energizado, se esta lâmpada se apagar, possivelmente teremos um trip da planta.

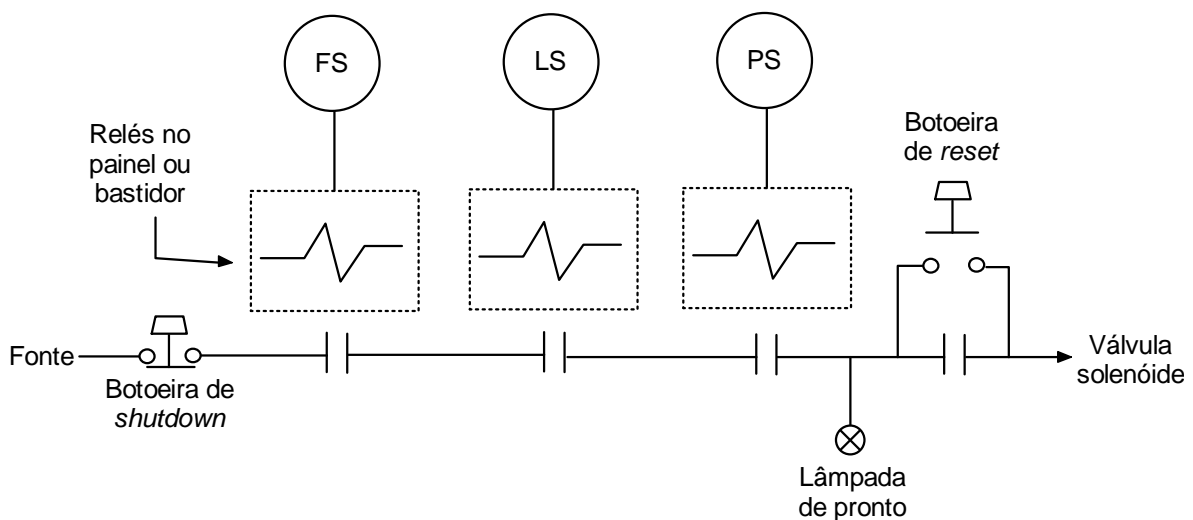


FIGURA 13 – SISTEMA A RELE TÍPICO.

### 3.2.3. SIS de Lógica Fixa em Estado Sólido

Os sistemas de lógica fixa foram concebidos para substituir os reles com menor dimensão, menores circuitos de potência (por exemplo, CMOS: *Complementary Metal Oxide Semiconductor*), estes sistemas são muito especiais, são relativamente caros, tem aplicação limitada, e, como resultado, já não são tão comuns.

Estes dispositivos são montados em forma de módulos, cartões que implementam uma determinada lógica, ou seja, são cartões com a possibilidade de ser configurada uma lógica de acordo com a sua necessidade, esta “programação” é feita na parte de trás do *hack* onde ficam conectados os módulos lógicos, ela é feita por fios.

### Vantagens:

- Distribuído pela planta;
- Alta confiança, em alguns casos chegando até SIL 4;
- Disponibilidade de comunicação serial;
- Disponibilidade de diagnóstico;
- Sem causa comum;
- Falha em modo seguro;
- Intrinsecamente seguro para Zona 2;
- Alta imunidade eletromagnética.

### Desvantagens:

- Pouco flexível;
- Documentação;
- Custo;
- Alguns modelos sem reposição hoje em dia.

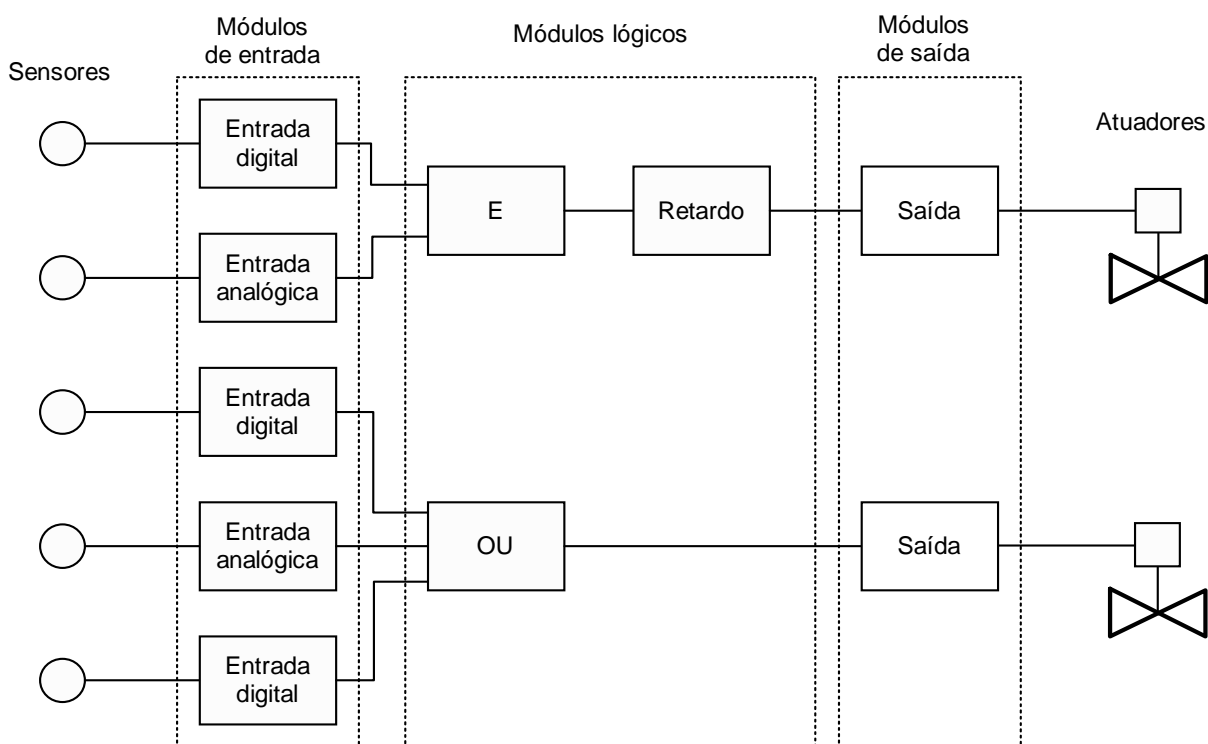


FIGURA 14 – CONCEITO DE OPERAÇÃO.

Conforme ilustrado na Figura 14, os sensores são ligados aos módulos de entradas. Estes são ligados aos módulos lógicos através de uma configuração por fio, a saída do circuito lógico é ligada ao módulo de saída e estes estão ligados aos atuadores.

#### 3.2.4. SIS Microprocessado (CLP)

Softwares baseados em sistemas são utilizados mais frequentemente. Não há nenhum imperativo tecnológico, que diz que tenhamos de utilizar sistemas baseados em microprocessador. Controladores Lógicos Programáveis (CLPs) foram originalmente criados para substituir sistemas a reles. Hoje em dia, seria praticamente impossível não mencionar-se sobre CLP para utilização em sistemas de segurança.

##### Vantagens:

- Flexibilidade;
- Testes e diagnósticos;
- Temporizador *watch dog timer*;
- Comunicação Serial;
- Controle de acesso;
- Reprogramável via *software*;
- Interface gráfica de programação;
- Documentação.

##### Desvantagens:

- Dependência de software;
- Falhas com causa comum;
- Comunicação com outros dispositivos;
- Custo.

Com o passar do tempo foi verificada as limitações que os CLPs de propósito geral apresentavam em sistemas de segurança críticos, logo, algumas empresas

chegaram à conclusão que precisavam desenvolver sistemas mais sofisticados. Então, em meados de 1970, começaram a estudar sistemas tolerantes a falhas em computação.

Segundo a IEC 61508 o que difere um sistema de segurança de um sistema comum é o nível do diagnóstico, implementação da redundância e a certificação independente.

Verifica-se na tabela abaixo, os principais modelos de alguns fabricantes de CLP de segurança certificados IEC 61508, aonde é possível constatar a classificação SIL dos CLPs e a sua estrutura interna.

TABELA 4 – TABELA DE CLPs DE SEGURANÇA CERTIFICADOS IEC 61508

Empresa	Modelo	Classificação	Estrutura do Sistema	Agência Certificadora
ABB	Safeguard 400	SIL 2/ SIL 3	1oo1D Hot Standby/ 1oo2D *proc. comp.	TUV Sud
Emerson	Delta V SIS	SIL 3	1oo2D *proc. comp.	Exida
HIMA	A1 A1 dig	SIL1/ SIL 2	1oo1/ 1oo1D Hot Standby	TUV Rheinland
HIMA	H41 H51q	SIL 3	2oo4D/ 2x *proc comp 1oo2D *proc. comp.	TUV Sud
Honeywell	FSC	SIL 3	1oo2D *proc. comp.	TUV Sud
Siemens	S7-400FH	SIL 3	1oo2D *proc. comp.	TUV Sud
Siemens	S7-300F	SIL 2	1oo1D Hot Standby	TUV Sud
Triconex	Tricon V.9	SIL 3	2oo3 Redundância Modular Tripla (TMR)	TUV Rheinland
Yokogawa	Prosafe PLC	SIL 3	1oo2D *proc. comp.	TUV Sud

\* Processador complementar.

Averigua-se que a maioria destes CLPs atendem a um nível de integridade de segurança 3 (SIL 3), denotando que eles tem uma alta confiabilidade e conseqüentemente uma baixa probabilidade de falha sob demanda (PFD).

Mais informações sobre os sistemas de outros fabricantes e *download* dos certificados pode ser encontrado em TUV (2009).



Alguns fornecedores seguem a estrutura de módulos redundantes com diagnósticos. Em termos de confiabilidade estes sistemas tem o mesmo efeito de um sistema 2oo3, atendendo SIL 3. Esta estrutura tem circuitos que verificam cada parte do sistema, desde o de entrada até o de saída, se for detectada alguma falha em algum destes circuitos a saída de diagnóstico abre, podendo causar um trip, mas nesta configuração temos dois canais 2oo2 com diagnóstico, conforme apresentado na Figura 15.

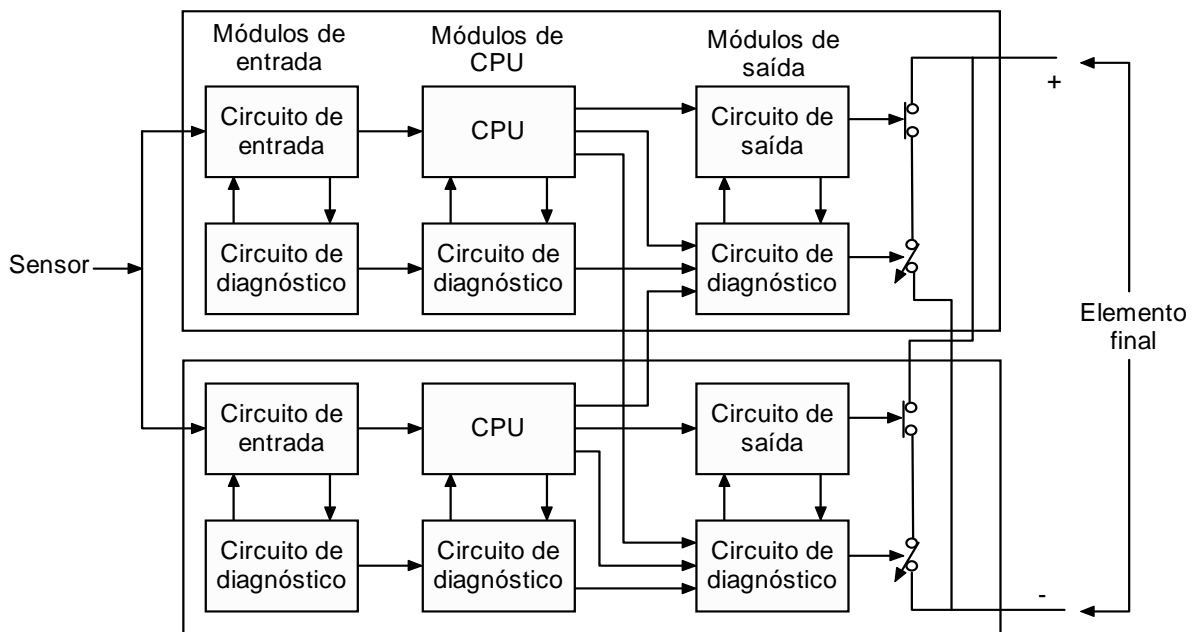


FIGURA 15 – SISTEMA 2oo2 COM DIAGNÓSTICO.

### 3.3. PROPOSTA DE UM SISTEMA TOLERANTE A FALHAS

A tolerância à falhas é uma das qualidades que um controlador pode apresentar. É a capacidade de detectar transitório e o estado de equilíbrio do erro adotando medidas on-line corretivas.

O sistema proposto será desenvolvido baseado na arquitetura do TMR, o sistema consiste em três sistemas de arquitetura idêntica. Cada canal é isolado um do outro, nenhum ponto de falha de um canal pode passar para outro canal. Se uma falha de hardware ocorrer, o canal defeituoso é invalidado.

O diagrama de blocos da Figura 16 representa os canais do sistema que serão implementados. O primeiro canal é o de aquisição do sinal, o segundo é o de processamento, e o terceiro canal de é o de votação, ele é quem decide qual sinal irá ser aplicado no processo.

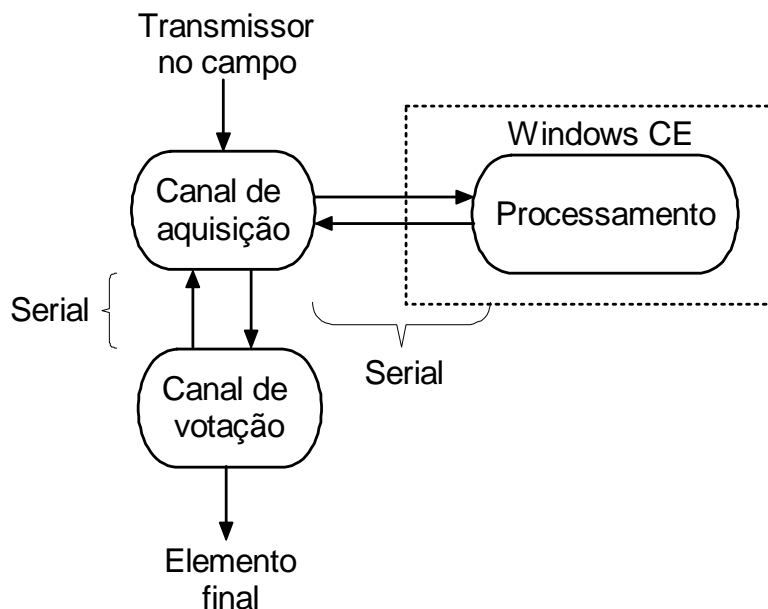


FIGURA 16 – DIAGRAMA DE BLOCOS DE UM CANAL DO SISTEMA PROPOSTO.

Desenvolver-se-á placas aquisição e transferência de sinal. Este circuito irá utilizar um microcontrolador PIC que estará lendo os dados de um transmissor de 4 à 20mA (transmissores analógicos) por um canal A/D de 10 bits e estará transferindo estes dados para os computadores (canais de processamento) via protocolo de comunicação serial RS-232 *full duplex*. É importante observar que cada placa de aquisição está recebendo o sinal de um transmissor diferente, e cada um está transmitindo o sinal para um computador diferente. Estes computadores processam estes sinais e devolvem os resultados para o canal de aquisição, se utilizando do mesmo protocolo serial, conforme apresentado na Figura 17.

Estes computadores (canais de processamento) farão o papel das CPUs, para tanto, será utilizando um sistema operacional de tempo real Windows CE. Após desenvolver-se a imagem no *Platform Builder*, dever-se-á configurar a imagem para *boot* local, isto deve ser feito para que o sistema possa operar sem ter um gerente de *boot*, pois o servidor de imagem limitaria o sistema como um todo.

Será desenvolvido dois *softwares* específicos, o primeiro será um algoritmo de controle que irá calcular os parâmetros do PID, este deverá ser testado controlando um processo real em uma planta didática, o segundo irá executar uma lógica de intertravamento, o qual será utilizado para levantar as taxas de falhas do sistema que serão compiladas para a realização do cálculo da confiabilidade de todo o sistema. Estes programas serão desenvolvidos em linguagem C, no compilador *Microsoft Embedded Visual C++*.

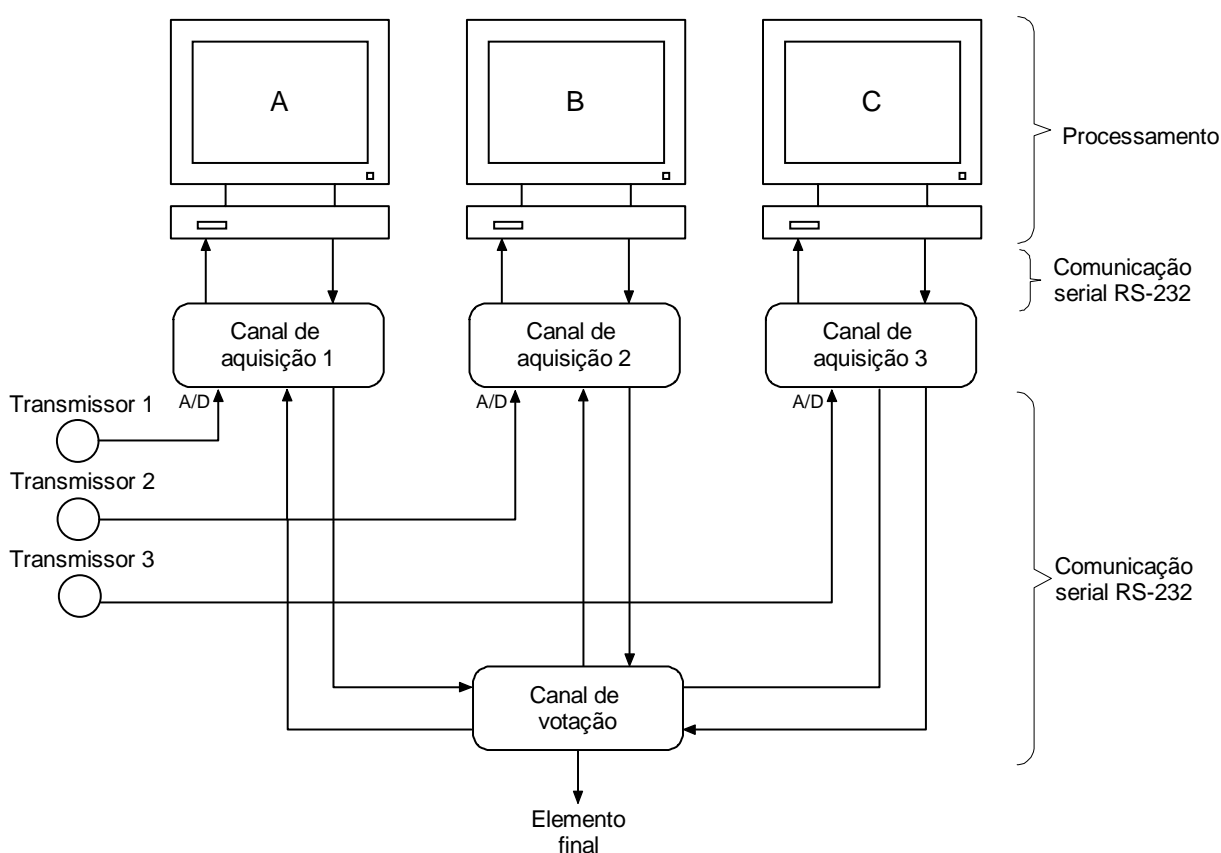


FIGURA 17 – DIAGRAMA DE INTERLIGAÇÃO.

Analisando a Figura 17, nota-se que serão utilizados transmissores analógicos, tanto para o algoritmo de controle como para o de intertravamento. Em um primeiro instante, isto parece errado, pois costumeiramente se utilizam sensores como chaves para sistemas de intertravamento, mas para sistemas de segurança *shutdown*, este artifício se torna interessante, pois o sensor está sendo testado continuamente, o que não aconteceria se fosse utilizado sensores tipo chave.

O canal de votação recebe os valores calculados pelo canal de processamento que foi enviado para o canal de aquisição. Através de uma

comunicação serial RS-232 *full duplex* ele recebe de cada canal o valor final calculado. Ele também identifica se algum canal não está em funcionamento. É o canal de votação quem decide qual é o valor que será aplicado no elemento final de controle, uma das formas de decisão é através do cálculo do valor mediano, ou seja, se dois computadores calcularem o valor da saída de controle para 50% e um calcular para 10%, ele assume a saída como sendo o canal que tem mais votações, nesse caso 50%.

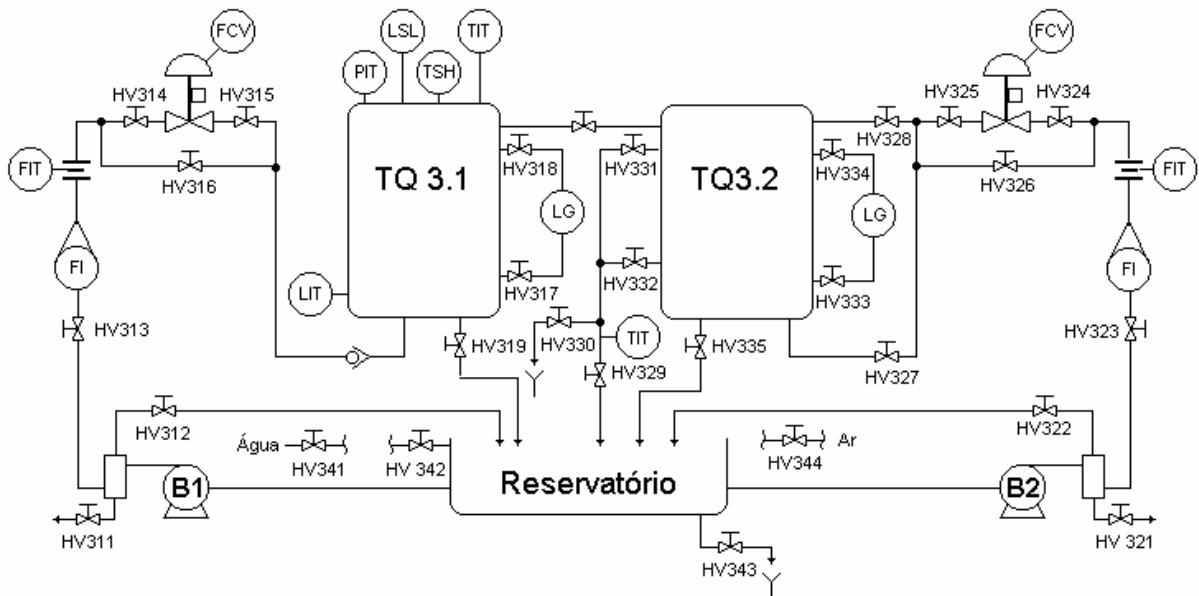
É o canal de votação que atua no circuito D/A do posicionador de válvula que recebe um sinal padronizado de 4 a 20mA, isto se este sistema for aplicado para controle, mas se for aplicado para *shutdown* à saída é um contato aberto ou fechado.

### 3.3.1. Planta Didática

A planta que será utilizada para a validação do controlador redundante proposto, é uma planta didática de controle de temperatura, nível e vazão, cujo nome dado é planta 03. O processo dela é produzir água em uma temperatura controlada. Todos os equipamentos são industriais, contém chaves de topo para fazer intertravamento de nível, visores de níveis nos dois tanques, termostato e pressostato no tanque 1, transmissor de temperatura nos dois tanques, indicadores de vazão do tipo rotâmetro, transmissor de nível, derivador de fluxo, posicionador de válvula eletrônico e eletro-pneumático e transmissor indicador de vazão, de acordo com o diagrama P&I apresentado na Figura 18.

O processo começa com o aquecimento da matéria prima (água) no tanque 3.1. Para que as termo resistências sejam acionadas é necessário que a chave de topo que faz o intertravamento de nível seja atuada. Quando esta chave atua a bomba para de mandar o líquido para o tanque 3.1 e o sistema de temperatura é acionado. Quando a temperatura chegar ao valor ajustado no termostato, o sistema de aquecimento é desligado e novamente inicia o bombeamento de água para o tanque 3.1, com a finalidade que a água aquecida passe para o tanque 3.2, onde será feito o controle de temperatura.

### Planta 03 - Controle de nível, vazão e temperatura











Área 31: Malha de vazão e temperatura do aquecimento do produto  
 Área 32 e 33: Malha de vazão e temperatura da mistura de produtos

FIGURA 18 – DIAGRAMA DA PLANTA 3.

A Tabela 5 relaciona os elementos que são utilizados na planta 3 descrevendo cada equipamento de acordo com a sua função no processo, conforme a norma ISA 5.1. A Tabela XX pode ser utilizada para a interpretação do Figura 18 e 19.

TABELA 5 – EQUIPAMENTOS E SUAS FUNÇÕES.

Equipamento	Função/descrição
	Válvula de retenção
	Bomba (Bomba 1)
	Válvula de acionamento manual tipo registro de gaveta
	Elemento primário para medição de vazão / Placa de orifício
	Válvula de acionamento automática/ deslocamento linear
	Transmissor indicador de vazão/ instrumento discreto instalado no campo
	Válvula de controle de vazão/ instrumento discreto instalado no campo

	Indicador de vazão/ instrumento discreto instalado no campo, rotâmetro
	Transmissor indicador de nível/ instrumento discreto instalado no campo
	Transmissor indicador de temperatura/ instrumento discreto instalado no campo
	Visor de nível/ instrumento discreto instalado no campo
	Transmissor indicador de pressão/ instrumento discreto instalado no campo
	Chave de nível baixo/ instrumento discreto instalado no campo
	Chave de temperatura alta/ instrumento discreto instalado no campo
	Válvula manual

No momento em que a água quente começa a entrar no tanque 3.2 o sistema de controle começa a trabalhar, pois o termostato do reservatório 3.1 foi ajustado para uma temperatura acima da temperatura de controle, desta forma, precisamos adicionar água fria para que ocorra o resfriamento e o controle. A temperatura é medida pelo transmissor de temperatura do tanque 3.2 e o controlador define qual é a quantidade de água que deve ser adicionada. Então a segunda malha de controle começa a controlar essa quantidade de mistura.

Esta seria uma das aplicações possíveis para se trabalhar com a planta 3. Para a aplicação em questão, terá que ocorrer algumas modificações na instrumentação desta planta, pois, se faz necessário três transmissores medindo a mesma variável. Para isto, utilizar-se-á apenas a primeira malha de controle, e irá trabalhar com o controle de nível. Cada transmissor irá medir o nível em sua tomada de pressão e irá transmitir este valor em um sinal de 4 à 20mA para o controlador.

Deverão ser colocados mais dois transmissores de nível no tanque 3.1 conforme o P&I proposto na Figura 19, visto que, já se tem um instalado. Estes três transmissores irão enviar um sinal de 4 à 20mA para o sistema redundante proposto que irá calcular a saída do controlador e irá realimentar um posicionador de válvula que está ligado a uma válvula de controle que controla a vazão do fluido que está indo para o tanque 3.1. Será feito um controle de nível pela vazão de água que está entrando no tanque 3.1.

## Diagrama Proposto

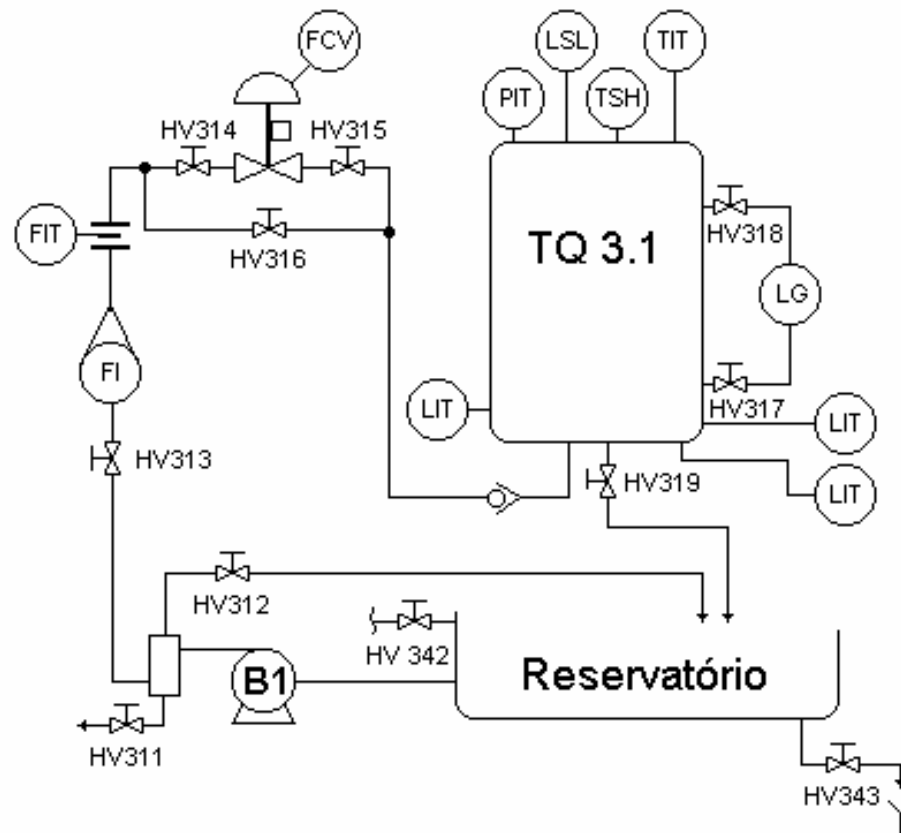


FIGURA 19 – DIAGRAMA PROPOSTO.

A *priori* será testado o sistema de controle e a *posteriori* será feito os testes para o cálculo da probabilidade de falha sobre a demanda (PFD), sistema de *shutdown*.

### 3.3.2. Comparação do Sistema com CLP Industrial de Controle

Os resultados obtidos do sistema de controle redundante serão comparados com os dados de um CLP de controle industrial, neste caso usar-se-á o LC 700 da Smar o qual já se encontra instalado na planta 3. Estes dados serão apresentados em forma gráfica no *software Matlab*. Serão inseridas algumas falhas sistemáticas, e os resultados obtidos serão discutidos e comparados. Para cada sistema de controle, o redundante e o industrial, será feita uma sintonia de controle respeitando

a estrutura de cada algoritmo de controle (PID), o método de sintonia utilizado será o da tentativa sistemática.

### 3.3.3. Obtenção de Dados para o Cálculo da PFD

Para a obtenção das taxas de falhas do sistema, será desenvolvido um sistema que ficará simulando modos de falhas no canal de aquisição de dados. Este mesmo sistema ficará verificando o comportamento da saída. Ele irá simular uma falha e verificar se a saída respondeu como o esperado, senão ele analisa qual foi o tipo de falha que ocorreu no sistema, se foi falha segura ou falha perigosa. Após o ciclo de testes, estes dados serão apresentados em um display LCD (*Liquid Crystal Display*) 16x2 para que seja feita a coleta dos mesmos.

Este sistema de simulação irá analisar a resposta para mil ciclos de entrada, cada ciclo será gerado após um segundo, ou seja, serão necessários 16,67 minutos para realizar uma varredura de mil ciclos.

Após estes dados serem coletados e tabulados, será possível aplicar o método descrito na norma IEC 61508 para o cálculo da PFD de sistemas redundantes.

## 3.4. AQUISIÇÃO DA PFD

Segundo a IEC 61508, o cálculo da PFD para sistemas redundantes pode ser obtido através do emprego Análise da Árvore de Falhas (*Fault Tree*). Para a aplicação deste método é necessário conhecer as taxas de falhas de cada circuito envolvido na arquitetura do sistema.

Na Figura 20(a) é demonstrado o circuito de votação de uma arquitetura 2oo3, as saídas são ligadas em série e paralelo. Avaliando-se a Figura 20(b), falha segura, percebe-se que se faz necessário que duas saídas falhem de modo seguro para todo o sistema falhar e olhando a Figura 20(c), falha perigosa, observa-se que



é necessário que duas saídas fiquem curto-circuitadas (falha perigosa) para que ocorra uma falha perigosa no sistema.

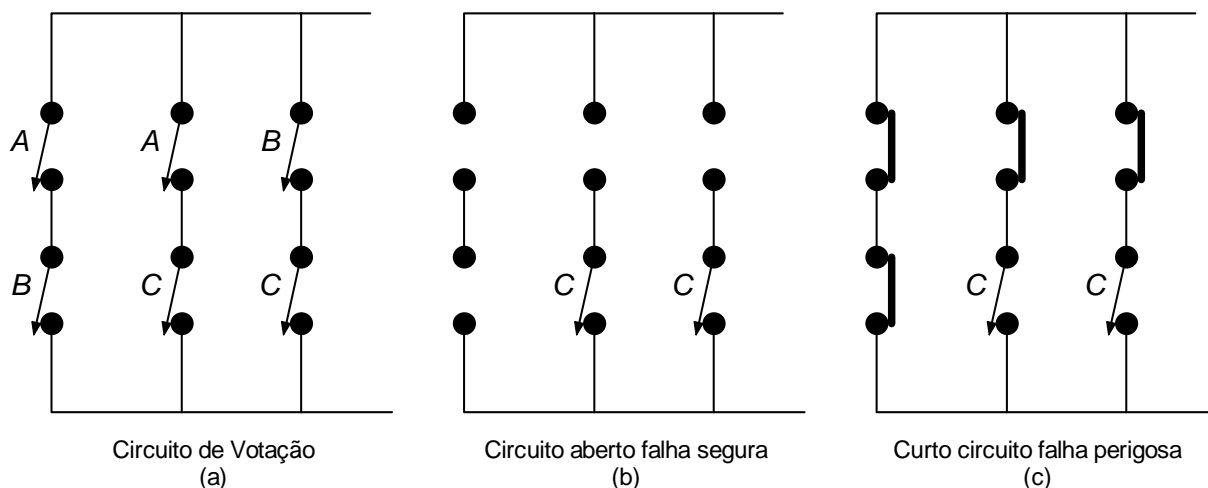


FIGURA 20 – MODOS DE FALHA DA ARQUITETURA 2oo3.

O sistema pode falhar perigosamente se ocorrer às falhas nos canais conforme a árvore de falhas indicada na Figura 21. Nesta Figura analisam-se os canais A e B, mas como o sistema é 2oo3 esta mesma árvore de falha é válida para os canais A e C, e para os canais B e C.

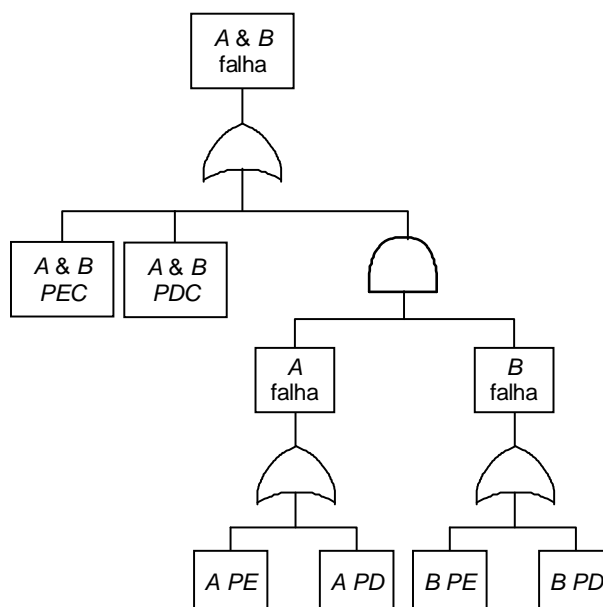


FIGURA 21 – ÁRVORE DE FALHA DE UMA ARQUITETURA 2oo3 PARA OS CANAIS A e B.

Considerando a árvore da Figura 21, conclui-se que os canais  $A$  e  $B$  irão falhar se ocorrer uma falha perigosa encoberta de causa comum ( $PEC$ ) em  $A$  e  $B$  ou se ocorrer uma falha de perigosa não descoberta de causa comum ( $PDC$ ) em  $A$  e  $B$ . Cada canal poderá falhar independentemente se ocorrer uma falha perigosa encoberta ( $PE$ ) ou uma falha perigosa descoberta ( $PD$ ) falhando assim os dois canais  $A$  e  $B$ . Logo a  $PFDA\&B$  é dada da seguinte maneira:

$$PFDA\&B = (A \& B)^{PEC} * RT + (A \& B)^{PDC} * TI + \left[ (A^{PD} * TI) + (A^{PEC} * RT) \right] \& \left[ (B^{PD} * TI) + (B^{PE} * RT) \right]$$

onde  $TI$  é o intervalo de testes e  $RT$  é o tempo de reparo.

Considerando que os canais são iguais, irá ser considerada a mesma taxa de falha  $\lambda$  para cada canal, reduzindo a fórmula anterior em:

$$PFDA\&B = (\lambda^{PEC} * RT) + (\lambda^{PDC} * TI) + \left[ (\lambda^{PD} * TI) + (\lambda^{PE} * RT) \right]^2. \quad (15)$$

A Fórmula 15 é para obter a  $PFDA\&B$  relacionada ao canal  $A$  e  $B$  como o sistema proposto é 2oo3 é necessário aplicar esta equação para as três combinações ( $A$  e  $B$ ,  $A$  e  $C$ ,  $C$  e  $B$ ), calculando-se então a  $PFDA\&B$  do sistema da seguinte forma:

$$PFDA\&B = 3(\lambda^{PEC} * RT) + 3(\lambda^{PDC} * TI) + 3 \left[ (\lambda^{PD} * TI) + (\lambda^{PE} * RT) \right]^2. \quad (16)$$

## CAPÍTULO 4

### 4. IMPLEMENTAÇÃO E RESULTADOS

Neste capítulo serão apresentadas duas aplicações deste sistema redundante, o primeiro será uma aplicação em um sistema em controle de nível em um processo industrial e a segunda em uma configuração de CLP de *shutdown*, intertravamento de segurança. Nestas aplicações serão detalhados os aspectos de construção de cada canal, bem como, os fluxogramas dos algoritmos que foram implementados.

Como se trata de um sistema redundante tem-se canais triplicados, é o caso do canal de aquisição e de processamento. O canal de votação não é necessário ser redundante, pois é ele quem vai calcular a mediana dos resultados finais para a aplicação do sinal no elemento final de controle.

#### 4.1. CANAL DE AQUISIÇÃO

Conforme demonstrado na Figura 17, o transmissor está ligado no canal de aquisição, este envia um sinal de 4 a 20mA para um conversor de corrente tensão gerando uma tensão de 1 a 5 V. Esta tensão então é aplicada em um pino A/D do microcontrolador PIC 16F876A (conforme pode ser visto no Apêndice A).

O algoritmo inicia as duas comunicações seriais que serão utilizadas ambas para 19200bps, inicia também o conversor analógico digital (A/D) e uma fonte de saída de dados (*Display* de LCD). Enquanto não ocorre nenhuma interrupção serial ele fica lendo a *PV* (variável do processo) e para que ocorra um amortecimento maior deste sinal e para que uma possível oscilação não venha a dar uma variação na resposta do sistema de controle este algoritmo lê 10 vezes o sinal do conversor A/D e calcula a média deste, assumindo como sinal final o resultado da média, conforme apresentado na Figura 22.

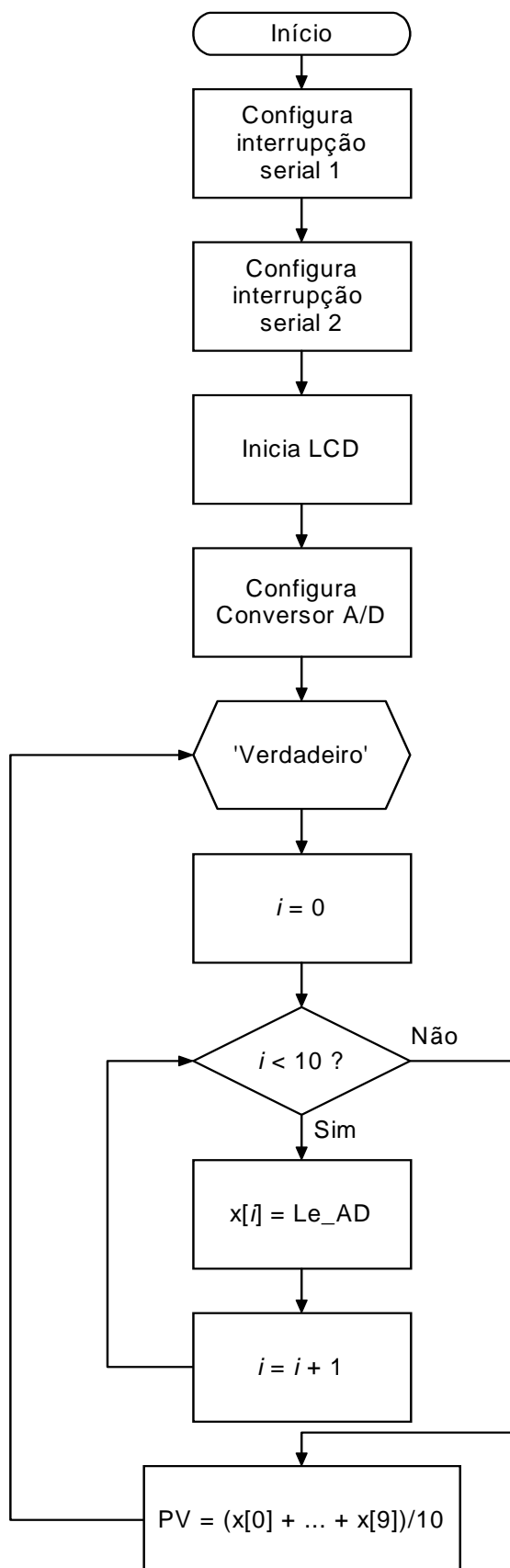


FIGURA 22 – FLUXOGRAMA DO ALGORITMO DO CANAL DE AQUISIÇÃO.

Este algoritmo tem duas fontes de interrupção serial, a primeira é para realizar a comunicação com o sistema embarcado no computador e a segunda é para que este canal se comunique com o canal de votação. A segunda fonte de interrupção desempenha duas funções, uma para enviar o valor do *CO* (saída de controle) para o canal de votação calcular a mediana e a outra é para que o canal de votação consiga identificar se o canal de aquisição contém algum defeito.

Quando ocorrer a interrupção serial 1 o programa vai para a função de tratamento da interrupção da serial 1 significando que o sistema embarcado está perguntando para o canal de aquisição qual é o valor da *PV* atual, então ele envia esse valor e aguarda a transmissão do novo valor da *CO*, como mostrado na Figura 23.

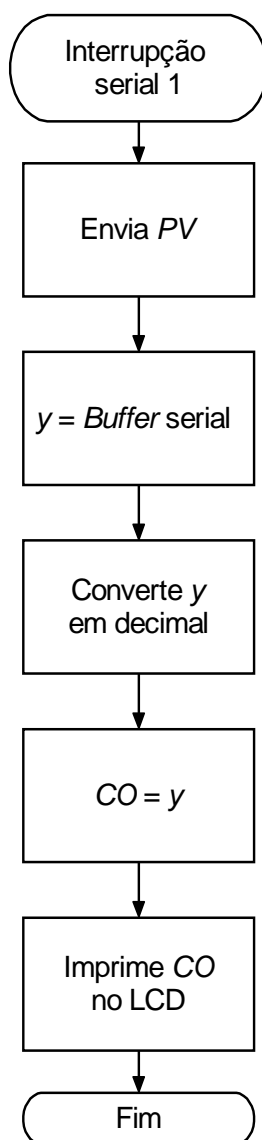


FIGURA 23 – FLUXOGRAMA DA FUNÇÃO DE TRATAMENTO DA INTERRUPÇÃO SERIAL 1.

Quando ocorrer a interrupção serial 2 o programa vai para a função de tratamento da interrupção da serial 2 significando que o canal de votação está perguntando para o canal de aquisição qual é o valor da CO atual, então ele envia esse valor e volta para onde estava antes de ocorrer esta interrupção.

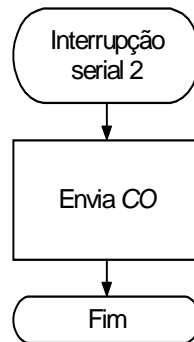


FIGURA 24 - FLUXOGRAMA DA FUNÇÃO DE TRATAMENTO DA INTERRUPÇÃO SERIAL 2.

Estas são as funções que o canal de aquisição está desenvolvendo, lê a variável do processo, transmite para o controlador, recebe a nova variável da saída do controlador e guarda este dado até que seja requerido pelo canal de votação.

#### 4.2. CANAL DE VOTAÇÃO

Todo o sistema depende do correto funcionamento do canal de votação, é ele que vai fazer o sincronismo do sistema. A sua principal função é o cálculo da mediana dos valores no CO que será transformado em um sinal de 4 a 20 mA e será modulado em um posicionador de válvula, atuando diretamente no elemento final de controle. Para tanto, usou-se um sinal PWM, onde com o valor do CO calcula-se a porcentagem do *duty-cycle*. Este sinal é aplicado a um conversor de tensão para corrente ativo, para que não exista a queda de 0,6V de tensão na junção base emissor de um transistor (este circuito pode ser visto no Apêndice B).

No caso da utilização deste sistema para *shutdown* é ele quem executa o algoritmo 2oo3 que decide se o elemento final deve ser fechado ou permanecer aberto. Conforme mostrado na Figura 20 (a). Para isto, foi utilizado um

microcontrolador PIC 16F876A implementando uma comunicação serial e a configuração de uma saída PWM.

De acordo com a função que o sistema terá no processo, controle ou *shutdown*, deverá ser utilizado o algoritmo correspondente. Desta forma, desenvolveram-se dois programas distintos, mas esta diferença está localizada diretamente no sistema de votação: o primeiro faz a votação da mediana para saber qual é o valor mais votado nos três sistemas o que ganhar será assumido como valor padrão de saída do elemento final e o segundo é o algoritmo de decisão de um sistema de votação 2oo3.

No fluxograma de controle inicialmente o algoritmo configura o PWM, configura o LCD e configura a comunicação serial para uma velocidade de 19200 bps. Então é feita uma varredura de cada canal. Após a verificação do canal de aquisição, o canal de varredura aguarda o recebimento da nova variável via serial, se isto não ocorrer o canal é dado como defeituoso. Se o canal responder corretamente este valor é assumido como o novo valor da variável temporária, novo CO. Estes dados são colocados em uma variável temporária, pois na próxima varredura estas variáveis podem assumir um outro valor.

Com as variáveis atualizadas no canal de votação o algoritmo realiza o cálculo da mediana dos valores, verificando se existe dois ou mais valores iguais, se isto ocorre assume-se o valor da saída do controlador como o valor que mais se coincide, caso isto não ocorra, ou seja, as três variáveis sejam diferentes, o novo valor que é ajustado o PWM é 0%.

Após o cálculo da variável CO, PWM\_atual, o *duty cycle* do PWM é ajustado para este novo valor. Através da conversão deste valor para um sinal de corrente atua-se no elemento final, posicionador de válvula de controle. Nas Figuras 25 e 26 apresentam-se os dois fluxogramas, o primeiro é o de controle e o segundo é o de intertravamento de *shutdown*.

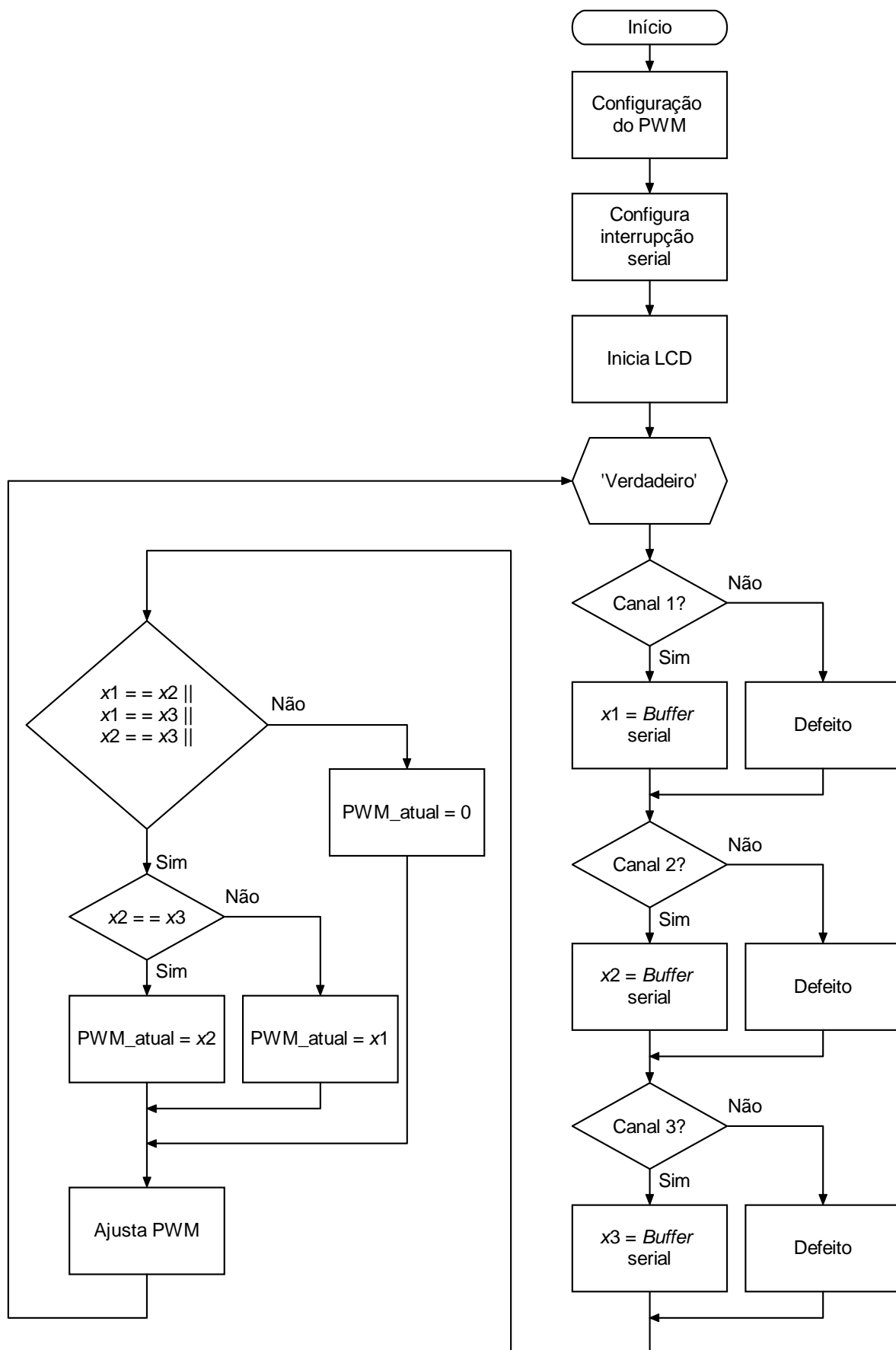


FIGURA 25 – FLUXOGRAMA DO ALGORITMO DE CONTROLE DO CANAL DE VOTAÇÃO.



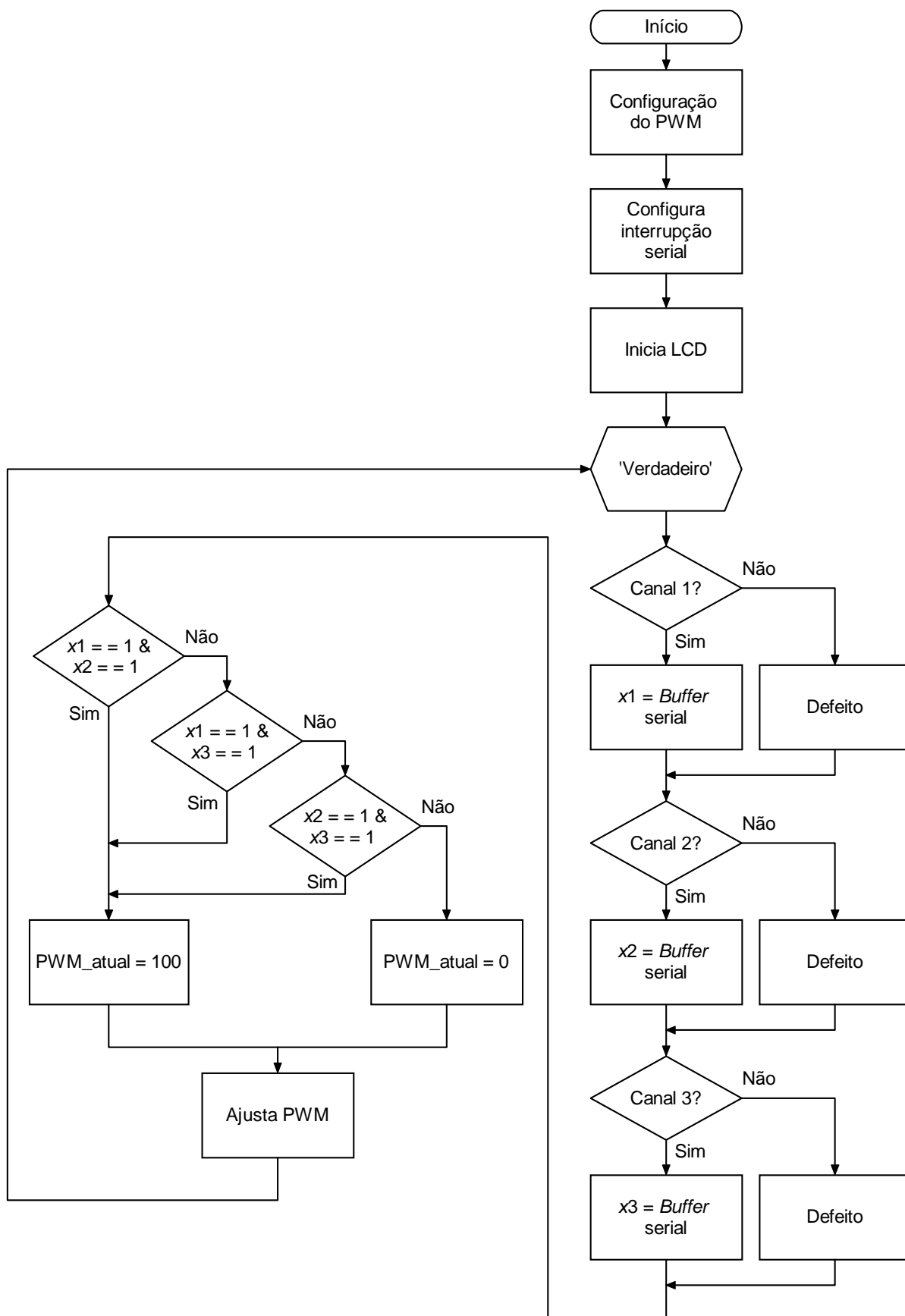


FIGURA 26 – FLUXOGRAMA DO ALGORITMO DE *SHUTDOWN* DO CANAL DE VOTAÇÃO.

Este algoritmo faz uma análise de sensores tipo chave, ou seja, 0 ou 100% aberto ou fechado, uma combinação binária, logo, ele se difere do outro justamente na análise da combinação de chaves e na saída resultante que é uma válvula 100% aberta ou totalmente fechada (depende da ação da válvula). Mas no algoritmo de controle precisamos encontrar o valor que se localiza dentro do range de 0 a 100%.

### 4.3. CANAL DE PROCESSAMENTO

Este é o canal onde irá ocorrer o processamento da informação, o cálculo do CO considerando o ajuste das variáveis de controle do algoritmo PID, a execução da lógica de intertravamento indicando se a saída deve ser aberta ou fechada considerando os estados de entrada e etc.

Para tanto, após ter sido instalada a plataforma do Windows CE chamada de *Platform Builder*, foi necessário criar a imagem que iria rodar na memória RAM dos computadores com o Windows Embarcado. Feita a imagem foi preciso configurá-la para que o *BOOT* do sistema operacional fosse local, pois neste projeto não teremos um servidor de *BOOT*, o que não faria sentido nesta aplicação, para isso foi preciso criar um disco de *BOOT*, inicialmente. Executado todos estes passos iniciou-se então o desenvolvimento dos sistemas utilizando o compilador *Microsoft Embedded Visual C++*.

#### 4.3.1. Criando a Imagem do Windows CE

A imagem construída tem um conjunto de características, estas definem os elementos a serem integrados, os *device drivers* utilizados, as configurações de hardware, entre outros.

O conjunto completo de características é apresentado na Tabela 6.

TABELA 6 – CONJUNTO DE CARACTERÍSTICAS UTILIZADAS PARA A CONSTRUÇÃO DA IMAGEM NO WINDOWS CE.

<i>Board Support Packages (BSPs)</i>	<ul style="list-style-type: none"> <li>• <i>CEPC:X86</i></li> </ul>
<i>Plataform Configuration</i>	<ul style="list-style-type: none"> <li>• <i>Custom Configuration</i></li> </ul>
<i>Custom Device</i>	<ul style="list-style-type: none"> <li>• <i>Custom Device with Shell and Graphical User Interface (GUI)</i></li> </ul>
<i>Aplications and Services Development</i>	<ul style="list-style-type: none"> <li>• <i>Active Template Library</i></li> <li>• <i>C Libraries and Runtimes</i></li> <li>• <i>Microsoft Foundation Classes (MFC)</i></li> <li>• <i>Standard SDK for Windows CE .NET</i></li> <li>• <i>.Net Compact Framework</i></li> </ul>
<i>Aplications - End User</i>	<ul style="list-style-type: none"> <li>• <i>Cab Files Installer/Uninstaller</i></li> <li>• <i>File Viewers</i></li> <li>• <i>Remote Desktop Connection</i></li> </ul>
<i>Core OS Services</i>	<ul style="list-style-type: none"> <li>• <i>Battery Driver</i></li> <li>• <i>Serial Port Support</i></li> <li>• <i>Parallel Port Support</i></li> <li>• <i>USB Host Support</i></li> <li>• <i>Debugging Tools</i></li> <li>• <i>Power Management</i></li> <li>• <i>Kernel Features</i></li> </ul>
<i>Communication Services and Networking</i>	<ul style="list-style-type: none"> <li>• <i>Networking – Local Area Network</i></li> </ul>
<i>File Systems and Data Store</i>	<ul style="list-style-type: none"> <li>• <i>File System – Internal</i></li> <li>• <i>Registry Storage</i></li> <li>• <i>Storage Manager</i></li> </ul>
<i>Fonts</i>	<ul style="list-style-type: none"> <li>• <i>Arial</i></li> <li>• <i>New Times Roman</i></li> </ul>
<i>Internet Client Services</i>	<ul style="list-style-type: none"> <li>• <i>Internet Explorer 6.0 for Windows CE Components</i></li> </ul>
<i>Multimedia Technologies</i>	<ul style="list-style-type: none"> <li>• <i>Audio</i></li> </ul>
<i>Shell and User Interface</i>	<ul style="list-style-type: none"> <li>• <i>Shell</i></li> <li>• <i>User Interface</i></li> </ul>

Depois desta plataforma pronta deve-se fazer a compilação da mesma, esse procedimento deve ser feito selecionando *build plataform* no menu *build*. *Drivers* e aplicativos podem ser adicionados à imagem depois da imagem construída, mas uma nova compilação deve ser realizada.

É importante salientar que foi construída apenas uma imagem para as duas aplicações, o que altera é somente o software que está sendo executado no sistema embarcado.

#### 4.3.2. Criando o Disco de *BOOT*

Acessar um programa chamado *Websetup.exe* que se encontra numa pasta da instalação do *Platform builder*, um exemplo do caminho que pode se encontrar esse arquivo é: C:\Arquivos de programas\Windows CE Platform Builder\4.20\cepb\utilities.

Depois de instalado o Programa *WebImage*, na mesma pasta que se encontra o executável *Websetup*, encontra-se um arquivo chamado *Cepcboot*, este arquivo é responsável pela criação de um disco de boot, ou seja, um disco responsável por iniciar a máquina onde é feito o *download* da imagem do sistema operacional.

#### 4.3.3. Efetuando o *BOOT* Local

Quando foi criada a imagem o *Platform Builder* gerou um arquivo com todos os *driver* e aplicativos selecionados chamado de *NK.bin*, este arquivo é copiado para o *Hard Disc* onde será efetuado o *boot* juntamente com todos os arquivos criados no disco de inicialização. A partir deste momento o *boot* já pode ser dado localmente. Possivelmente a resolução inicial seja de 640x480, para corrigir isto, basta editar o arquivo *autoexec.bat* e alterar a resolução manualmente para a desejada. É muito importante que o programa *Loadcepc.exe* seja copiado para o *Hard Disc*, pois é ele quem chama o arquivo *NK.bin*

#### 4.3.4. Criando um Novo Projeto

Após a construção da imagem no *Platform Builder* e as configurações de *boot* local finalizadas iniciou-se o desenvolvimento do primeiro aplicativo. Para tanto, utilizou-se o compilador *Microsoft Embedded C++ 4.0*. Para que fosse possível a utilização deste compilador foi necessário atualizar o *servicepack* para a versão 3. Por algum motivo este compilador não cria aplicativos MFC (*Microsoft Foundation*

*Classes*) senão estiver atualizado o *servicepack* para versão 3. Este arquivo pode ser encontrado para *download* em Microsoft (2009).

Após a realização destes passos, abre-se o compilador e cria-se um novo projeto, na lista de projetos seleciona-se a opção *WCE MFC AppWizard(exe)* e na lista de CPUs seleciona-se Win32 (WCE x86). Na criação do projeto é importante selecionar a opção *Windows Sockets*.

#### 4.4. SISTEMA DE CONTROLE

Uma das aplicações dos sistemas redundantes na indústria é o controle de processos que apresentam grandes risco as pessoas da operação, a fábrica e a comunidade. Um exemplo desse tipo de processo é o controle de caldeira, as caldeiras da Petrobrás REPAR em Araucária geram vapor saturado a uma pressão de aproximadamente  $88\text{Kgf/cm}^2$  a uma temperatura próxima de  $405^{\circ}\text{C}$ , logo, este é um processo que necessita de um sistema de controle tolerante a falhas devido ao perigo que ele representa para empresa e para a comunidade que vive ao seu redor.

O sistema de controle proposto executa um algoritmo que calcula as ações de controle de um controlador do tipo PID. Na aplicação desenvolvida ele precisa fazer o controle de nível de uma planta. Este programa de controle está sendo executado na plataforma Windows CE, recebendo a *PV* (variável do processo) do canal de aquisição via serial, executando o algoritmo PID e enviando, via serial, o *CO* (saída de controle) para a placa de aquisição, como descrito anteriormente.

Toda a lógica do algoritmo do cálculo do PID, valor de ajuste do *SP* (set-point) está programado neste sistema. É ele quem faz a pergunta do valor da *PV* para o canal de aquisição em um tempo pré-definido.

O fluxograma da Figura 27 ilustra como está estruturada a troca de dados entre o sistema de controle e o canal de aquisição.

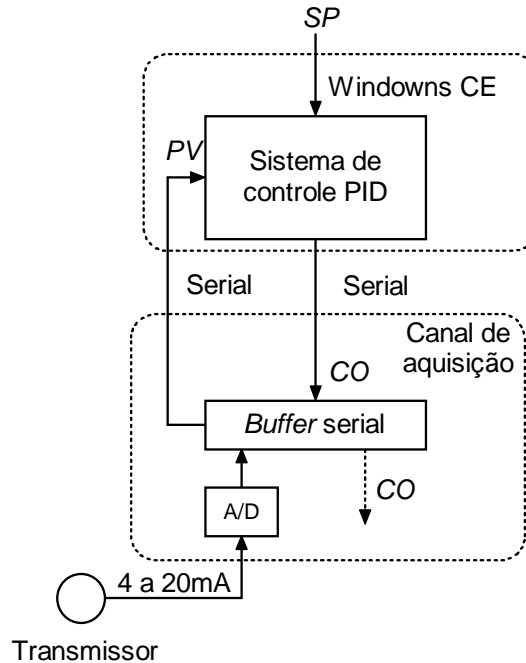


FIGURA 27 – FLUXOGRAMA DA TROCA DE DADOS ENTRE O CANAL DE AQUISIÇÃO E O WINDOWS CE.

No sistema de controle tem-se duas entradas de dados a  $PV$  que vem do canal de aquisição e o  $SP$  que é definido pelo usuário, o erro é calculado internamente subtraindo a  $PV$  do  $SP$  ( $Erro = PV - SP$ ). Ainda existem mais três variáveis de entrada que são as ações de controle:  $Kp$  (ganho proporcional),  $Ti$  (tempo integral) e  $Td$  (tempo derivativo). Após o cálculo do PID o resultado de saída que é a variável  $CO$  é enviada para o canal de aquisição, via comunicação serial.

Este sistema de controle pode estar operando em manual ou em automático. Em manual significa que o sistema está em malha aberta e o ajuste do elemento final está sendo realizado pelo operador, em automático fecha-se a malha e agora quem controla todo o processo é o algoritmo PID. O fluxograma do algoritmo do sistema de controle pode ser visto na Figura 28.

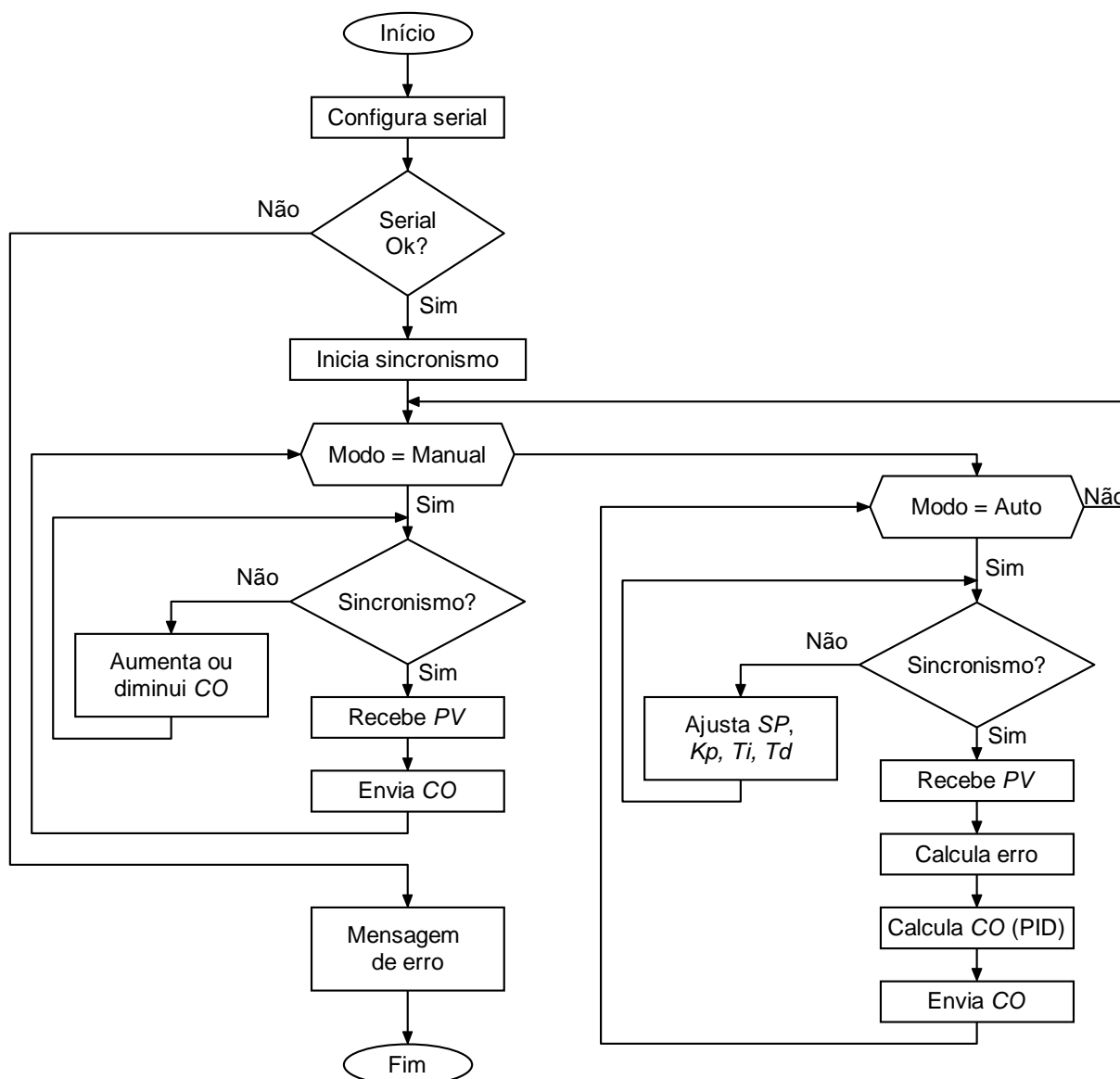


FIGURA 28 – FLUXOGRAMA DO SISTEMA DE CONTROLE.

O primeiro teste realizado pelo algoritmo é a verificação da comunicação serial, se por algum motivo esta esteja inoperante, o algoritmo devolve uma mensagem de erro de comunicação e este é finalizado.

Se o teste da comunicação passar, ajusta-se então um relógio que gera o sincronismo do sistema de controle com o canal de aquisição. O programa pode operar em modo manual e automático como dito anteriormente, se for configurado para modo manual (configuração inicial, devido à necessidade de ser feito o *start-up* no processo) o programa fica esperando o estouro do temporizador, enquanto isso pode ser ajustado o valor do CO, quando ocorrer o estouro do temporizador o

sistema recebe do canal de aquisição o valor da *PV* atual e envia o novo valor da variável *CO*.

A partir do momento em que o sistema é passado para automático, o usuário não tem mais direito a modificar a saída de controle, ele passa a ter que ajustar o *SP* e as variáveis de controle, agora todo o controle (o cálculo do novo *CO*) depende do algoritmo PID que recebe o valor da *PV* atual em seguida calcula o erro subtraindo a *SP* da *PV*, calcula o novo *CO* e o envia para o canal de aquisição.

Antes de sair controlando o processo, levantou-se a curva de resposta do mesmo. Para isto passou o sistema para malha aberta e o ganho proporcional ( $K_p$ ) foi ajustado para um ganho unitário. O sistema foi levado até uma faixa de operação e aguardou-se que este entrasse em um regime permanente, feito isto, aplicou-se um sinal degrau de 10% de amplitude e a partir deste momento foi feita a aquisição destes dados, com isto, conseguiu-se levantar a curva de resposta deste sistema em malha aberta, como pode ser visualizado na Figura 29, este gráfico foi plotado no *Matlab*.

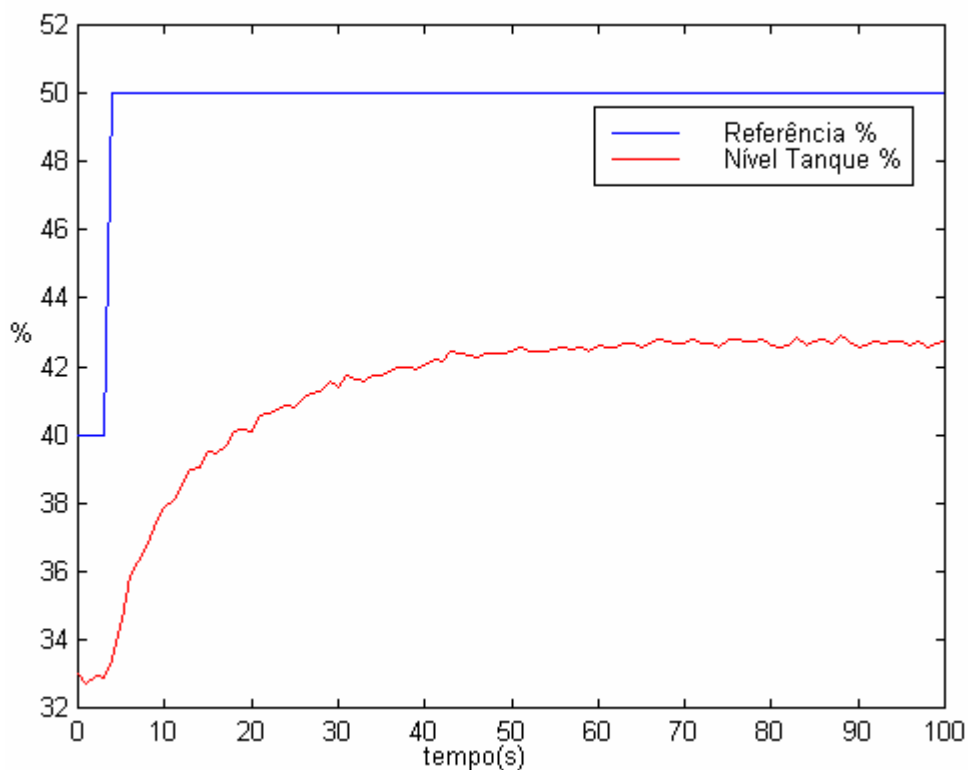


FIGURA 29 – CURVA DE RESPOSTA DO SISTEMA EM MALHA ABERTA.



Após a obtenção da curva de resposta do processo em malha aberta, aplicou-se o primeiro método de *Ziegler e Nichols*, conhecido como método da curva de resposta em malha aberta, para obter a função de transferência do processo. Este método pode ser aplicado a plantas que não envolvam integradores nem pólos complexos dominantes. A função de transferência pode ser aproximada por um sistema de primeira ordem com tempo morto (atraso de transporte). A equação 15 apresenta a função de transferência da planta didática que será controlada pelo sistema proposto e pelo CLP industrial, dada por:

$$G(S) = \frac{1,65}{10S + 1}. \quad (15)$$

Com a Figura 29 de resposta do processo, iniciou-se o processo de sintonia do controlador, o primeiro a ser sintonizado foi o sistema que está sendo proposto neste trabalho de dissertação. Como é o mesmo algoritmo que está sendo executado nos três canais de processamento, as ações de controle foram setadas para os mesmos valores para que fosse possível ter um mesmo valor de sintonia nos três controladores.

Como esta curva de resposta caracteriza um sistema de primeira ordem, foi utilizado um controlador do tipo PI, ajustou-se o  $K_p$  para 4 e o  $T_i$  para 1 e como não foi usado o  $T_d$  este ficou em 0, uma forma de ser anulada esta ação de controle. Para gerar a curva de resposta do processo sendo controlado, aplicou-se um degrau de amplitude igual a 10% no  $SP$ . O gráfico de resposta do sistema controlado pode ser visualizado na Figura 30.

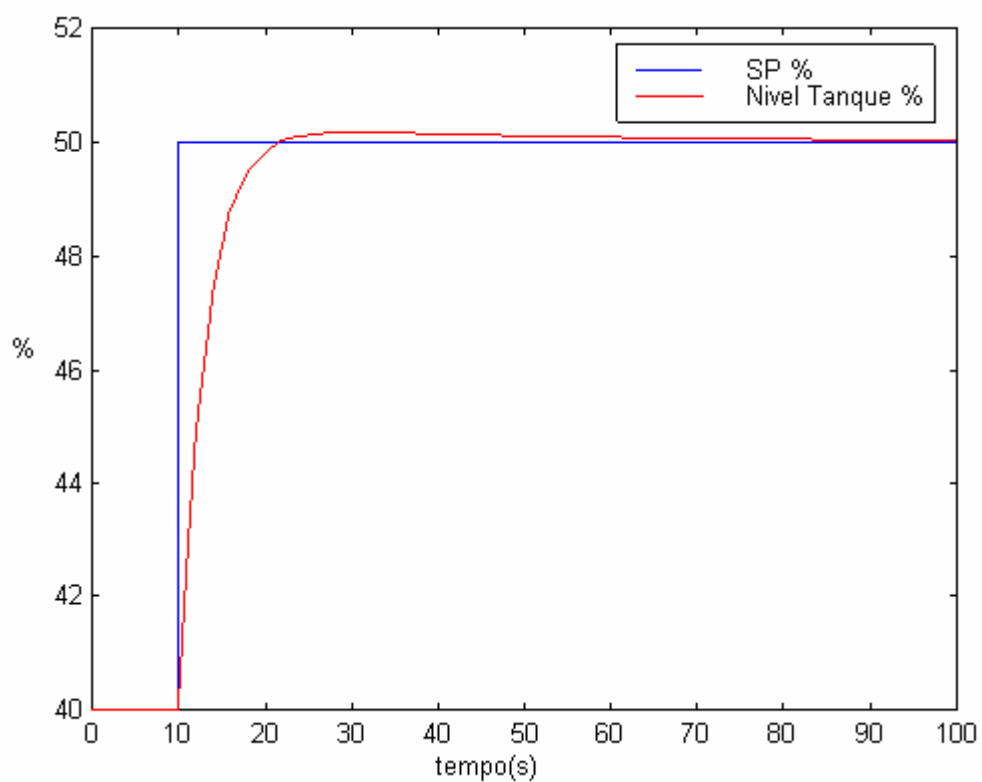


FIGURA 30 – CURVA DE RESPOSTA DO PROCESSO CONTROLADO PELO SISTEMA PROPOSTO.

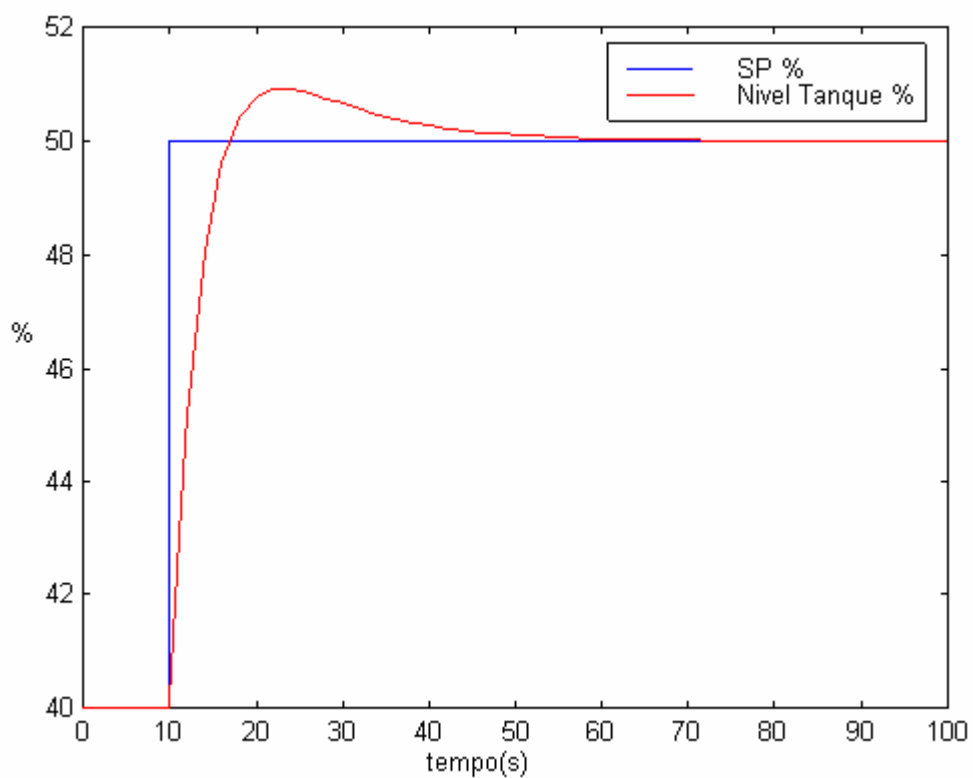


FIGURA 31 – CURVA DE RESPOSTA DO PROCESSO CONTROLADO POR UM CONTROLADOR INDUSTRIAL.

Para que fosse feita uma comparação de resposta entre sistemas, utilizou-se um CLP industrial o LC-700 da Smar como controlador do mesmo processo, e este foi sintonizado e submetido aos mesmos testes do sistema proposto nesta dissertação.

A Figura 31 apresenta a resposta obtida do controle realizado pelo LC-700. Verifica-se que existe uma diferença de resposta nos dois gráficos, mas isto se deve a estrutura do PID e a sintonia do controlador. No processo em questão esta diferença não é significativa, logo, assumem-se os dois controles como satisfatório, o que é muito positivo para este estudo, pois se conseguiu demonstrar que o sistema redundante proposto tem uma resposta em controle tão boa quanto um sistema industrial.

Foi efetuada a simulação de queda de alimentação, para tanto, desligou-se uma fase de alimentação de um canal de processamento, como o sistema é redundante e tem três CPU à falta de uma fase não influenciou o desempenho do sistema, o que não ocorreu com o LC-700, pois este é um sistema simples sem redundância quando foi desligada a fase de alimentação todo o sistema foi desligado inclusive todos os transmissores, neste momento o processo ficou sem controle.

#### 4.5. SISTEMA DE INTERTRAVAMENTO

Para a utilização do sistema proposto como sistema de intertravamento foi necessário realizar algumas modificações nos circuitos de aquisição, pois nesta aplicação se fez necessária a utilização de três entradas em cada canal de aquisição porque o sistema de processamento irá executar uma lógica booleana de três chaves (sensores *on-off*) aberto ou fechada.

O canal de votação estará verificando a resposta dos canais de processamento e decidirá se a saída estará acionada ou desacionada conforme a lógica do sistema de votação 2oo3.

Um sistema de simulação de falhas e defeitos foi desenvolvido para que se possa gerar falhas perigosas na entrada do canal de aquisição. Este sistema de simulação irá receber a saída do canal de votação e irá verificar se o sistema tomou

a decisão correta. Estes dados são mostrados em um display de LCD. O diagrama esquemático pode ser visto na Figura 32.

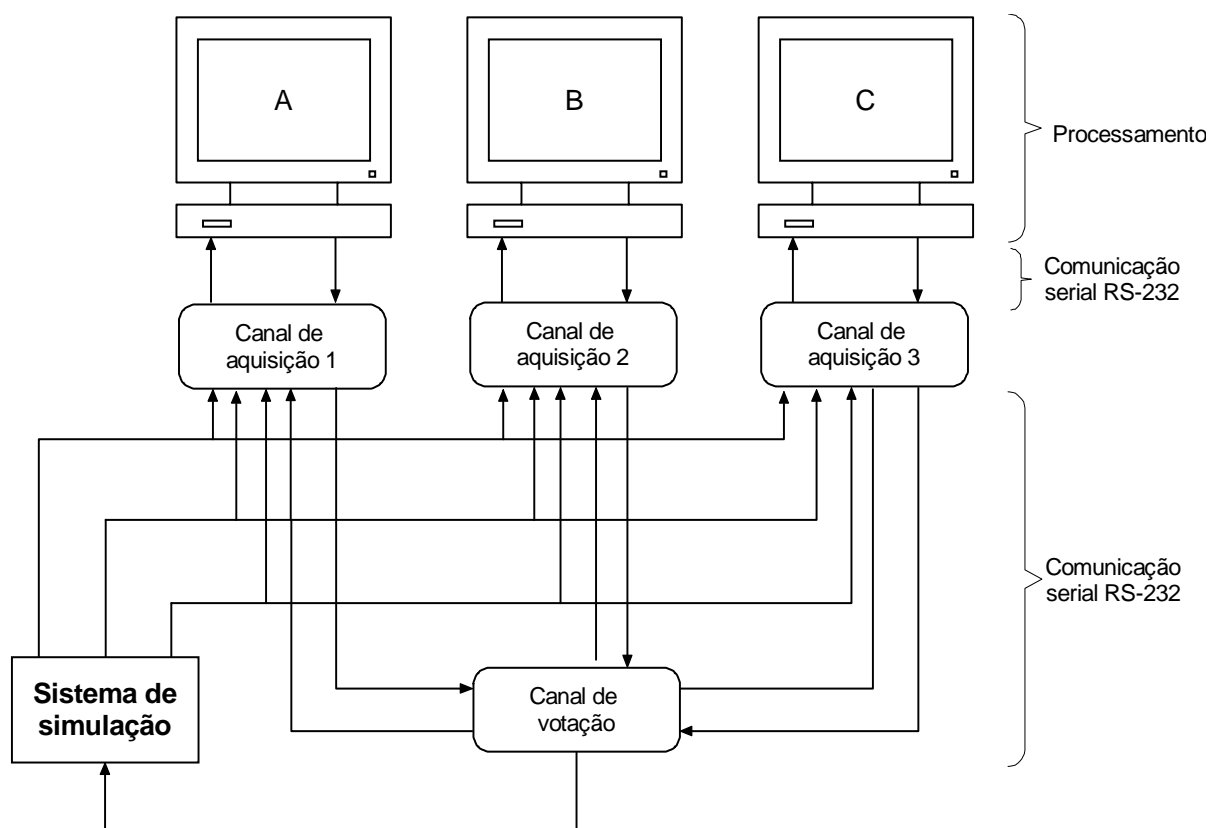


FIGURA 32 – DIAGRAMA ESQUEMÁTICO DO SISTEMA 2003.

A lógica que está implementada em cada canal de processamento é  $X = SA$  e  $SB$  e  $SC$ , ou seja, se alguma chave (sensor) enviar um sinal de nível lógico '0' a saída do canal de processamento deve ir para nível lógico '0', executando simplesmente a lógica booleana das entradas. Conforme a Tabela 7.

TABELA 7 – TABELA VERDADE DE SAÍDA DE CADA CANAL DE PROCESSAMENTO DADA AS ENTRADAS (SA,SB e SC)

SA	SB	SC	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

O sistema de simulação vai setando as variáveis SA, SB e SC para os valores da Tabela 7 e analisa a resposta do canal de votação. Por exemplo: setando SA para nível lógico '0', SB para nível lógico '1' e SC para nível lógico '1' a saída dos canais de processamento devem ser igual ao nível lógico '0', o que fará com que a saída do canal de votação vá para nível lógico '0', isto significa que o sistema está em perfeita operação, um sensor de segurança atuou e como a lógica nos três canais de processamento é a mesma todos responderam de forma igual e por consequência disso a resposta do sistema de votação foi efetuar o *shutdown* da malha. Mas se por algum motivo (falha), dadas as entradas anteriores, dois canais de processamento mantivessem a saída em nível lógico '1', o canal de votação não irá realizar o *shutdown* caracterizando desta forma uma falha perigosa, pois é necessário que a malha seja desligada mas o sistema não o fará. Da mesma forma isto pode ser avaliado se os sensores estiverem em nível lógico '1' e por algum motivo (falha) dois canais de processamento forem para nível lógico '0', isto fará com que o canal de votação vá para nível lógico '0' efetuando o *shutdown* da malha quando isto não deveria ocorrer, caracterizando assim uma falha segura.

O fluxograma do sistema de intertamento pode ser visto na Figura 33.

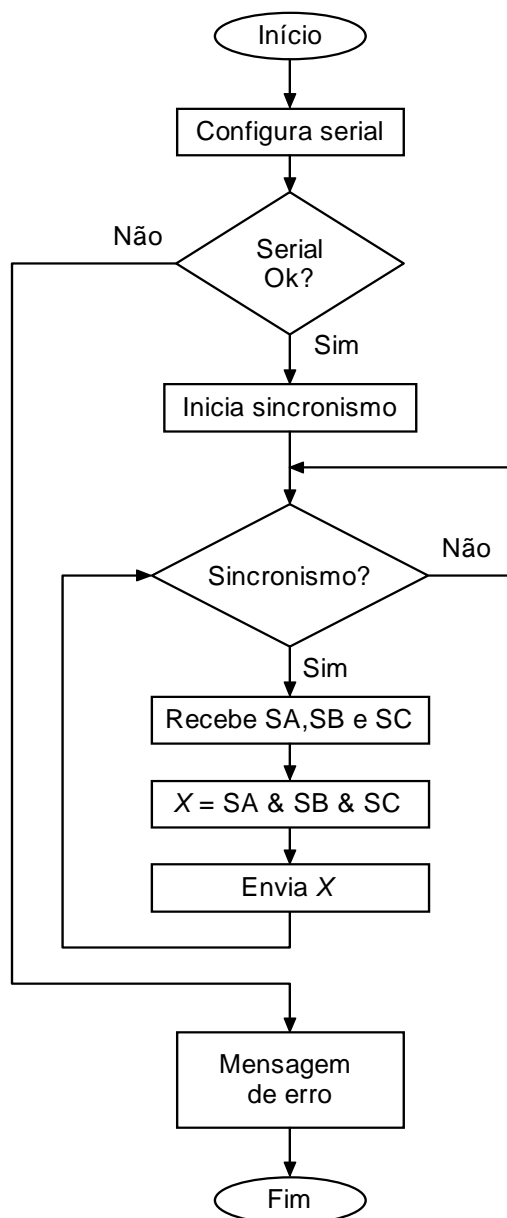


FIGURA 33 – FLUXOGRAMA DO SISTEMA DE INTERTRAVAMENTO.

Após as implementações e modificações tanto de *hardware* como de *software* iniciou-se os ciclos de testes para a obtenção dos dados para que fosse possível o cálculo da PFD, bem como, a disponibilidade do sistema proposto.

Para a coleta da falha segura foram feitas dez varreduras de mil ciclos cada, ao fim de cada varredura anotou-se o número de falhas seguras detectadas e calculou-se a média das falhas, gerando a Tabela 8.

TABELA 8 – FALHAS SEGURAS

Varredura	Ciclos	Falha segura ( $\lambda_s$ )
1	1000	145
2	1000	101
3	1000	95
4	1000	70
5	1000	46
6	1000	58
7	1000	51
8	1000	31
9	1000	48
10	1000	12
	Média de Falhas Seguras	65,7

$$\lambda_s = \frac{65,7}{1000} = 0,0657.$$

$$\sigma = 38,6955.$$

$$MTTR = 6 \text{ horas.}$$

Aplicando a norma IEC 61508 vem:

$$MTTF_{sp} = \frac{1}{6 * (\lambda_s)^2 * MTTR}$$

$$MTTF_{sp} = \frac{1}{6 * (0,0657)^2 * \left(\frac{6}{8760}\right)}$$

$$MTTF_{sp} = 56.373 \text{ anos.}$$

Após ser aplicada a fórmula de cálculo do  $MTTF_{sp}$  encontrou-se um tempo médio entre falhas de 56.373 anos o que significa que pode ocorrer uma falha em 56.373 anos neste sistema proposto. O  $MTTR$  utilizado foi de 6 horas, esse valor surge do tempo médio de reparo que foi utilizado durante os testes para o levantamento da taxa de falhas do sistema.

Para a coleta da falha perigosa foram feitas dez varreduras de mil ciclos cada, ao fim de cada varredura anotou-se o número de falhas perigosas detectadas e calculou-se a média das falhas, gerando a Tabela 9.

TABELA 9 – FALHAS PERIGOSAS

Varredura	Ciclos	Falha perigosa ( $\lambda_d$ )
1	1000	83
2	1000	71
3	1000	85
4	1000	70
5	1000	46
6	1000	48
7	1000	52
8	1000	46
9	1000	33
10	1000	12
Média de Falhas Perigosas		54,6

$$\lambda_d = \frac{54,6}{1000} = 0,0546.$$

$$\sigma = 22,8920.$$

$$TI = 1 \text{ ano.}$$

Aplicando a norma IEC 61508 vem:

$$PFD = (\lambda_d)^2 * (TI)^2$$

$$PFD = (0,0546)^2 * \left(\frac{1 \text{ ano}}{2}\right)^2$$

$$PFD = 0,000745$$

$$RRF = \frac{1}{(0,000745)} = 1341,76$$

$$D = 1 - 0,000745 = 0,999255 = 99,92547\%.$$

Os cálculos indicam que o sistema proposto tem um tempo médio entre “trips” de 56.373 anos, e uma disponibilidade de 99,92%, isto daria um nível de integridade SIL 3, de acordo com a norma IEC 61508 indicando que este sistema pode ser aplicado em malhas até SIL 3. O que é um resultado bem satisfatório, pois consegue-se visualizar que este sistema esta respondendo de acordo com o esperado, e esta demonstrado ter uma confiabilidade aceitável para o nível 3 de integridade de segurança.



## CAPÍTULO 5

### 5. CONCLUSÕES E TRABALHOS FUTUROS

No dia a dia da indústria se faz mais do que necessário à utilização de sistemas tolerantes a falhas, devido ao perigo e a complexidade que alguns processos apresentam para a indústria e para seus funcionários.

A tolerância à falhas é uma das qualidades que um controlador pode apresentar. É a capacidade de detectar transitório e o estado de equilíbrio do erro adotando medidas *on-line* corretivas.

Investir em sistemas extremamente caros não é sinônimo de confiabilidade, uma vez que as válvulas ainda constituem o elo mais fraco do sistema. É preciso ter em mente que uma malha de *shutdown* depende dos controladores, bem como, dos sensores e atuadores (válvulas).

Como foi demonstrado, existem várias opções de sistema de redundância no mercado, mas existem também malhas mal avaliadas, ou seja, super dimensionadas ou ainda mal dimensionadas encarecendo o projeto do SIS. As normas vigentes orientam no que deve ser feito em um SIS e não como.

Neste trabalho foi proposta uma arquitetura de redundância tolerante a falhas com a utilização do *hardware* do PC padrão. Também foi utilizado um sistema embarcado de tempo real, ou seja, foi retirado o sistema operacional padrão e foi colocado um sistema de tempo real, no caso Windows CE. Para que fosse possível a redundância foi instalado em três computadores o mesmo sistema embarcado e estes executaram o mesmo programa em paralelo.

As placas de circuito impresso, chamados de canal, foram desenvolvidas para que o PC tivesse acesso as variáveis de controle e a elementos finais de controle, para tanto, utilizou-se microcontroladores PIC que implementam conversores A/D de 10 bits.

O primeiro teste efetuado foi comparar este sistema com um CLP industrial, onde se obteve um resultado bem satisfatório, pode-se observar o real funcionamento de um sistema de redundância e a vantagem em relação a um sistema simplex.

O segundo teste foi fazer o levantamento das taxas de falhas, tanto as seguras como as perigosas, do sistema implementado e assim aplicar a norma IEC 61508 para calcular a disponibilidade do sistema e conseguir descobrir qual era o nível de integridade deste sistema, como resultado obteve-se SIL 3. Isto significa que o sistema proposto atende a malhas que exigem um nível de segurança 3. O que é um resultado excelente, pois todos os PES estudados atendem até SIL 3, a IEC 61508 é muito cautelosa com sistemas SIL 4.

A finalidade deste trabalho não é desmerecer os PES existentes, mas demonstrar que pode ser desenvolvido sistemas tão eficazes quanto aproveitando-se de *hardwares* existentes com modificações e evoluções dos *softwares*.

O Windows CE se demonstrou uma alternativa viável para o desenvolvimento de dispositivos dedicados, onde ele permite a seleção dos componentes do sistema operacional em uma vasta base de dados, garantindo total flexibilidade e um rápido desenvolvimento.

Além da aplicação industrial, acredita-se que este trabalho tem uma aplicação muito interessante no meio acadêmico, devido ao preço que um CLP de redundância apresenta. Uma das dificuldades encontradas, durante o desenvolvimento desta dissertação, foi encontrar uma empresa que estivesse disposta a emprestar o seu CLP de redundância para estudo, o que dificultou o desenvolvimento do mesmo. O sistema proposto pode ser utilizado em processos reais dentro de instituições de ensino como usinas piloto, controle de caldeiras e etc. A um custo bem inferior de um sistema industrial, mas tão eficaz quanto.

Entre os possíveis trabalhos futuros destacam-se:

- Produzir e disponibilizar literatura em português sobre Sistema de Intertravamento de Segurança, pois ainda existem poucos livros sobre o assunto no idioma português e considerando-se também outros tipos de bibliografia, como artigos;
- Desenvolver uma comunicação entre os três canais de processamento para que se tenha apenas uma interface com o usuário;
- Desenvolver este sistema utilizando a comunicação USB (*Universal Serial Bus*) entre os canais de processamento e de aquisição, para aumentar a velocidade na troca de dados aumentando assim o desempenho do sistema;

- Estudos de integração deste sistema com protocolos de redes industriais, como *Foundation Fieldbus* e *Profibus*;
- Pesquisar novas arquiteturas para o canal de votação, este é o “gargalo” do sistema de votação, utilizando-se talvez de sistemas inteligentes ou especialistas;
- Desenvolver sistemas embarcados para monitoramento local de válvula de controle, bem como, sistemas alternativos para realizar testes nestes elementos finais sem o desligamento do processo;
- Estudos de viabilidade econômica e de implantação deste sistema para aplicação industrial;
- Verificar o desempenho destes sistemas utilizando-se de outros sistemas embarcados, por exemplo, o *Linux Embedded*;
- Transformar esta dissertação em um produto viável, tanto na área acadêmica como no meio industrial.

Os interessados poderão dar continuidade à redação deste trabalho, afinal, esta dissertação, é uma obra embora não concludente, passível de revisão e ampliação.

## REFERÊNCIAS

- AVIZIENIS, A.; KELLY, J. P. **Fault tolerance by design diversity - concepts and experiments**. Computer, New York, USA, 1984.
- BARR, M. **Programming Embedded Systems in C and C++**. Sebastopol: O' Reilly & Associates, CA, 1999.
- BEGA, E. A. **Instrumentação Aplicada ao Controle de Caldeiras**. 3ª. ed. Rio de Janeiro, RJ: Interciência. 2003.
- BURNS, A.; WELLINGS, A. **Real-Time Systems and Programming Languages**. 2ª. ed. England: Addison Wesley, 1997.
- CENTER FOR CHEMICAL PROCESS SAFETY - CCPS. **Guidelines for Safe Automation of Chemical Processes**. New York: ISA, 1993.
- CHEN, L.; AVIZIENIS, A. N-version programming: a fault tolerance approach to reliability of software operation. In: **Annual International Symposium on Fault-Tolerant Computing**, 1978. *Proceedings*. New York, IEEE, 1978. p. 3-9.
- FINKEL V. S. Ferramentas modernas para avaliar os SIL e trabalhar o SIS das funções de segurança. **INTECH BRASIL**, São Paulo, n. 81, p. 8-14, 2006.
- FINKEL, V. S. *et al.* **Instrumentação Industrial**. 2ª. ed. Rio de Janeiro: Interciência, 2006.
- GOBLE, W. M., **Control Systems Safety Evaluation and Reliability**. 2ª. Ed. Research Triangle Park, USA: ISA, 1998.
- GOBLE, W. M.; CHEDDIE, H. **Safety Instrumented Systems Verification: Practical Probabilistic Calculations**. 1ª. ed. Research Triangle Park, USA: ISA, 2005.
- GRUHN, P.; CHEDDIE, H. **Safety Instrumented Systems: Design, Analysis, and Justification**. 2ª. ed. Research Triangle Park, USA : ISA, 2006.
- GUREWICH, N.; GUREWICH, O. **Visual C++ 5**. 4ª. ed. Indiana, USA: Sams, 1997.
- HORTON, I. **Visual C++ 2005**. 1ª. ed. Indianápolis, USA: Wiley Publishing, Inc, 2006.
- INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 61508: Functional Safety of electrical / electronic / programmable electronic safety-related systems**. Geneva: Switzerland, 2000.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 61511: Functional safety – Safety instrumented systems for the process industry sector**. Geneva: Switzerland, 2000.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 62061: Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems**. Geneva: Switzerland, 2005.

JAFFE, M.S.; LEVESON, N.G.; HEIMDAHL, M.P.E.M.; BONNIE, E. Software requirements analysis for real time process control systems. **IEEE Transactions on Software Engineering**, v.17, n.3, p.241-258, 1991.

JOHNSON, B.W. **Design and Analysis of Fault Tolerant Digital Systems**. University of Virginia, New York, USA: Addison-Wesley Publishing Company, 1989.

KITCHENHAM, B.; PELEEGER, S.L. Software Quality: The Elusive Target. **IEEE Software**, p.12-21, 1996.

LOURES, E. F. R. **VIEnCoD: Proposta de um Ambiente CACSD Baseado em Plataforma de Instrumentação Virtual e MATLAB**. Dissertação – Programa de Pós-Graduação em Informática Aplicada, Pontifícia Universidade Católica do Paraná, Curitiba, PR, 1999.

MICROSOFT. **Microsoft Corporation**, 2009. Real Time and Windows CE Disponível em: <[http://msdn.microsoft.com/embedded/usewinemb/ce/techno/realtime/default.aspx?\\_r=1](http://msdn.microsoft.com/embedded/usewinemb/ce/techno/realtime/default.aspx?_r=1)>  
Acesso em: 20/03/2009

MICROSOFT. **Microsoft Corporation**, 2009 eMbedded Visual C++ 4.0 SP4. Disponível em: <http://www.microsoft.com/downloads/details.aspx?FamilyID=4A4ED1F4-91D3-4DBE-986E-A812984318E5&displaylang=en>  
Acesso em: 07/04/2009

NISE, N. S. **Engenharia de Sistemas de Controle**. 3ª. ed. Rio de Janeiro, RJ: LTC, 2002.

OFFSHORE RELIABILITY. **OREDA: Offshore Reliability – Data Handbook**. 4ª. ed. Strindevieien, USA, 2002.

OGATA, K. **Engenharia de Controle Moderno**. 4ª. ed. Rio de Janeiro, RJ: Pearson, 2003.

OLIVEIRA, R.; CARISSIMI, A; TOSCANI, S. **Sistemas Operacionais**. 2ª. ed. Brasil: Sagra Luzzatto, 2001.

ORTIZ, S. JR. Embedded OS Gain the Inside Track. **IEEE Computer**. v. 34, n. 11, p. 14-16, 2001.

PEDROSO, J. de M. **Proposta de Utilização do Sistema Operacional Windows CE para aplicações didáticas na área de Automação e Controle.** 154 f. Dissertação – Setor de Tecnologia, Universidade Tecnológica Federal do Paraná, Curitiba, PR, 2007.

PEREIRA, F. **Microcontrolador PIC: Programação em C.** 4ª. ed. São Paulo, SP: Editora Érica, 2005.

PETROBRAS. **N-2595: Critérios de Projeto e Manutenção para Sistemas Instrumentados de Segurança em Unidades Industriais.** São Paulo, SP, 2002.

PETROBRAS. **N-2194: Especificação de Controladores Programáveis.** São Paulo, SP, 1996.

PHAM, H. **Handbook of Reliability Engineering.** 1ª. ed. New Jersey, USA: Springer, 2003.

PIAZZA, G. **Introdução à Engenharia da Confiabilidade.** 1ª. ed. Caxias do Sul, RS: EDUCS, 2000.

REESE, J. D.; LEVESON, N.G. Software Deviation Analysis: A “Safeware” Technique. **Annual Loss Prevention Symposium**, 31, p.1-14, Houston, Texas, USA, 1997.

SÁ, M. C. de. **Programação C para Microcontroladores 8051.** 1ª. ed. São Paulo, SP: Editora Érica, 2005.

SEAMAN, C. B. Qualitative Methods in Empirical Studies of Software Engineering. **IEEE Transactions on Software Engineering**, v. 25, n. 4, p. 557-572, 1999.

SMITH, D. J. **Reliability, Maintainability, and Risk: Practical Methods for Engineers.** 5ª. ed. Butterworth-Heinemann, 1997.

STOREY, N. **Safety-Critical Computer Systems.** New York, USA: Addison Wesley, 1996.

STROTHMAN, J. **Measurement Equations and Tables.** 2ª. ed. Research Triangle Park, USA: ISA, 2006.

TACKE, C.; RICCI, L. Benchmarking Real-Time Determinism in Windows CE. **White Paper** Disponível em: <<http://msdn2.microsoft.com/en-us/library/ms836535.aspx>>, 2002. Acesso em 12/03/2009.

THE INSTRUMENTATION, SYSTEMS, AND AUTOMATION SOCIETY. **ANSI/ISA TR-84.00.01: Functional Safety: Safety Instrumented Systems for the Process Industry Sector – Parts 1, 2, and 3,** Research Triangle Park, USA, ISA, 1996.

**TUV** - Cooperation Functional Safety. Germany, 2009. Disponível em: <<http://www.tuv-fs.com/plclist.htm>>. Acesso em: 27/03/2009.

UNIVERSIDADE FEDERAL DO PARANÁ. Sistema de Bibliotecas. **Referências Bibliográficas**. Curitiba: Editora UFPR, 2007. (Normas para apresentação de documentos científicos, 2ª. ed.).

US **MIL-HANDBOOK-217F** - Failure Rates for Electronic Components, USA, 1992.

XIE, M.; DAÍ, Y. S.; POH, K. L. **Computing Systems Reliability – models and analysis**. New York, USA: Kluwer Academic Publishers, 2004.

WEBER, R. F.; WEBER, T. S. Um experimento prático em programação diversitária. **III Simpósio em Sistemas de Computadores Tolerantes a Falhas, SCTF**. 20-22 set. Anais. Rio de Janeiro, RJ, 1989. p. 271-290.

ZIO, E. **An Introduction to the Basics of Reliability and Risk Analysis**. v.13. Singapore: World Scientific Publishing, 2007.

**ANEXO A**

**CERTIFICADO TUV DO TRICON DA TRICONEX**





**TÜV Rheinland Group**

**TÜV Industrie Service GmbH**  
Automation, Software und Informationstechnologie

# ZERTIFIKAT CERTIFICATE

Nr./No. 968/EZ 105.04/05

Prüfgegenstand Product tested	Safety Related Programmable Electronic System	Hersteller Manufacturer	TRICONEX Invensys Systems, Inc. 15345 Barranca Parkway USA-Irvine, California 92618 United States of America
Architektur Architecture	2oo3 with diagnostics (2oo3D) and 3-2-1-0 OR 3-2-0 configurable mode of operation		
Typbezeichnung Type designation	TRICON Version 10	Verwendungszweck Intended application	Safety Related Programmable Electronic System for process control, BMS, Fire and Gas, emergency shut down, where the safe state is the de-energized state.  Fire and Gas, where the demand state is the de-energized or energized state.
Prüfgrundlagen Codes and standards forming the basis of testing			IEC 61508, Part 1-7:2000 IEC 61511:2004 DIN VDE 0116:1989, EN 50156-1:2004 NFPA 85:2001 EN 61131-2:2003 EN 61000-6-2:2001, EN 61000-6-4:2001 EN 54-2:1997, NFPA 72:2002
Prüfungsergebnis Test results			The system is suitable for safety related applications up to SIL 3 considering the results of the test report-no. 968/EZ 105.04/05 dated 2005-08-15.
Besondere Bedingungen Specific requirements			For the use of the systems the test report mentioned above, the Safety Manual, the User Manual and the actual revision of the official list of product documentation, hardware modules and software components released by TRICONEX and TÜV Rheinland must be considered.



Der Prüfbericht-Nr. 968/EZ 105.04/05 vom 2005-08-15 ist Bestandteil dieses Zertifikates.

Der Inhaber eines für den Prüfgegenstand gültigen Genehmigungs-Ausweises ist berechtigt, die mit dem Prüfgegenstand übereinstimmenden Erzeugnisse mit dem abgebildeten Prüfzeichen zu versehen.

The test report-no. 968/EZ 105.04/05 dated 2005-08-15 is an integral part of this certificate.

The holder of a valid licence certificate for the product tested is authorized to affix the test mark shown opposite to products, which are identical with the product tested.

**TÜV Industrie Service GmbH**  
Geschäftsfeld ASI  
Automation, Software und Informationstechnologie  
Am Grauen Stein, 51105 Köln  
Postfach 91 09 51, 51101 Köln

*H. Gall*

2005-08-15

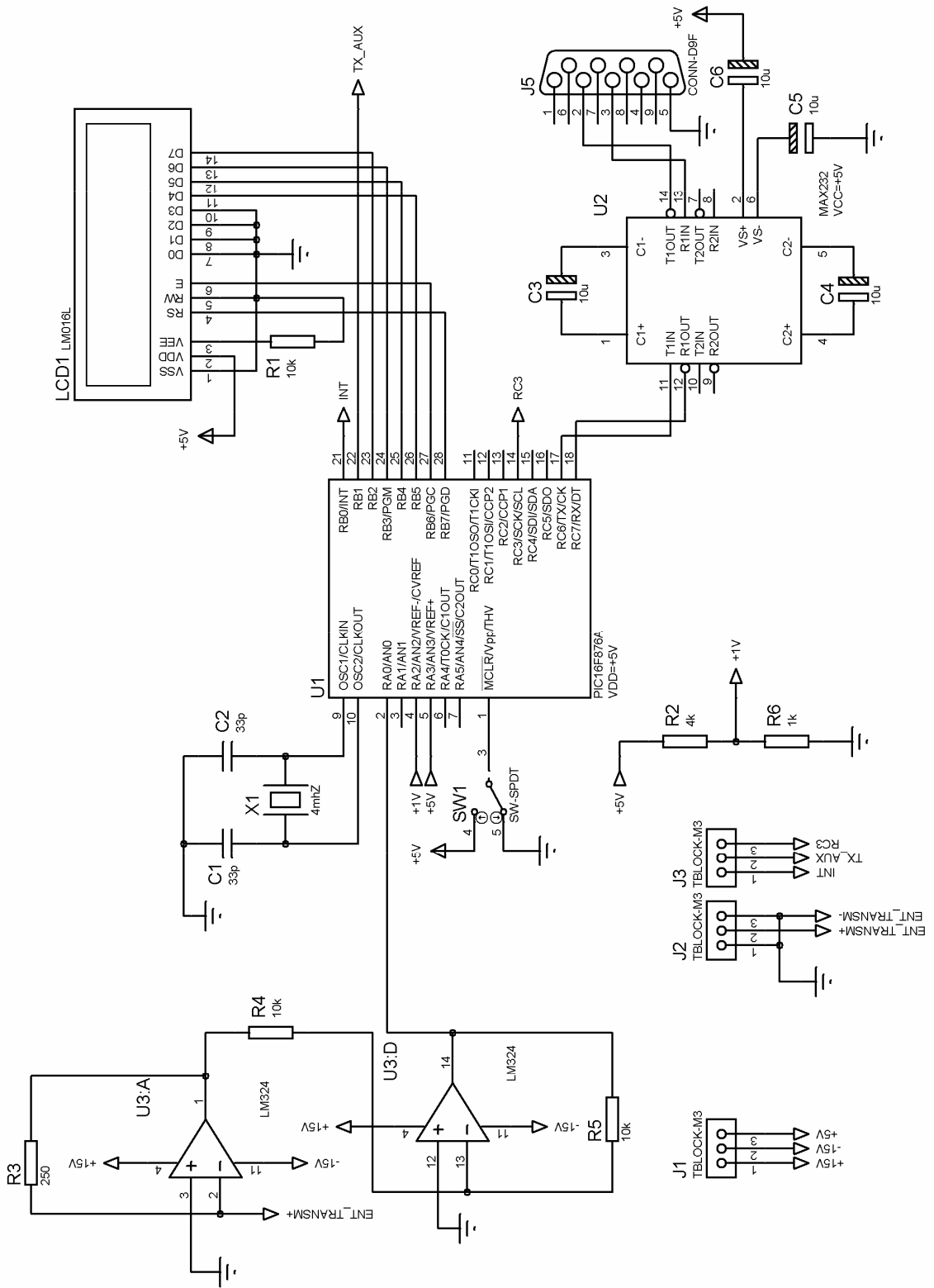
Datum/Date

Firmenstempel/Company seal

Dipl.-Ing. Heinz Gall

## APÊNDICE A

### CIRCUITO ELETRÔNICO DO CANAL DE AQUISIÇÃO



## APÊNDICE B

### CIRCUITO ELETRÔNICO DO CANAL DE VOTAÇÃO

