Serviço de Pós-Graduação EESC/USP

EXEMPLAR REVISADO

Data de entrada no Serviço: 10 / 02 / 2004

Ass.: Leandro Corcioli

**UMA ESTRATÉGIA EM REDES FIELDBUS USANDO
CONTROLE ADAPTATIVO POR MODELO DE
REFERÊNCIA APLICADA A SISTEMAS COMPLEXOS**

Elenilton Teodoro Domingues

DEDALUS - Acervo - EESC

Tese apresentada à Escola de Engenharia de
São Carlos, da Universidade de São Paulo,
como parte dos requisitos para obtenção do
título de Doutor em Engenharia Mecânica.

ORIENTADOR: Prof. Dr. Mario Pinotti Junior

São Carlos

2003

FOLHA DE JULGAMENTO

Candidato: Engenheiro **ELENILTON TEODORO DOMINGUES**

Tese defendida e julgada em 10-11-2003 perante a Comissão Julgadora:

Prof. Tit. **MARIO PINOTTI JUNIOR (Orientador)**
(Escola de Engenharia de São Carlos/USP) aprovado

Prof. Dr. **LUIS CARLOS PASSARINI**
(Escola de Engenharia de São Carlos/USP) aprovado

Prof. Dr. **MANOEL LUIZ DE AGUIAR**
(Escola de Engenharia de São Carlos/USP) aprovado

Prof. Dr. **EDILSON REIS RODRIGUES KATO**
(Universidade Federal de São Carlos/UFSCar) aprovado

Prof. Dr. **ROBERTO DE SOUZA JUNIOR**
(Universidade Metodista de Piracicaba/UNIMEP) aprovado

Prof. Associado **JONAS DE CARVALHO**
Coordenador do Programa de Pós-Graduação
em Engenharia Mecânica

Profa. Assoc. **MARIA DO CARMO CALIJURI**
Presidente da Comissão de Pós-Graduação

DEDICATÓRIA

*Aos meus pais, Felício e Aurea
com muito amor e carinho.*

Ao grande e soberano Deus, principal responsável pelo encorajamento na elaboração deste trabalho, cuja força e dedicação me foram concedidas durante toda esta etapa.

Em especial ao Prof. Doutor Mario Pinotti Junior, que através de sua experiência, me auxiliou e orientou sem medir esforços na elaboração deste trabalho.

Ao meu pai Felício que me ensinou a questionar e procurar o saber, à minha mãe Aurea que me ensinou a lutar e a dedicar-me a um ideal, ao meu irmão Edson e toda a minha família que me ensinaram o valor da união.

Especialmente gostaria de agradecer a minha esposa Ana Sabrina pelo carinho, amor, solidariedade, compreensão e incentivo nos momentos mais difíceis que encontrei.

Ao Capitão Cook e ao Grig pelas diversões e alegrias dos finais de semanas.

Aos professores Manoel e Passarini que sempre estiveram presente nesta caminhada.

Aos amigos, um dos tesouros da vida, que venho encontrando no caminho e que sempre me ajudaram e incentivaram.

Ao amigo Afonso Celso Pacheco pela amizade e força nos momentos difíceis.

Ao amigo Jaime Marini pelos conselhos e amizade.

Ao pessoal do Laboratório de Eletrônica Embarcada pela amizade.

A SMAR e sua equipe pela troca de informações.

À CAPES - Fundação Coordenação de Aperfeiçoamento de Pessoal de Superior, pela bolsa concedida.

Em muitas ocasiões não há como expressar o agradecimento merecido a todas as pessoas que me apoiaram, ajudaram e que com sua sabedoria me fizeram crer que vale a pena sonhar e lutar. Resta-me, portanto, agradecer de coração a todas elas.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	v
LISTA DE ABREVIATURAS E SIGLAS.....	vi
LISTA DE SÍMBOLOS.....	ix
RESUMO	xi
ABSTRACT	xii

1. INTRODUÇÃO	01
1.1 Introdução	01
1.2 Tecnologia de transmissão de dados e redes de campo	02
1.3 Malhas de controle com a tecnologia <i>Fieldbus</i> atual	06
1.4 A importância de algoritmos avançados de controle	08
1.5 Objetivo	09
1.6 Organização do trabalho	10
2. ESTADO DA ARTE – CONTROLE DE PROCESSOS	11
2.1 Introdução	11
2.2 Evolução tecnológica em controle de processo	12
2.2.1 Transmissão de sinais	12
2.2.2 Sistemas de transmissão de sinais	13
2.2.2.1 Transmissores inteligentes	13
2.2.2.2 Protocolos e padronização	14
2.2.2.2.1 Modelo de referência ISO/OSI	15

2.2.2.2.2 Modelo ISO/OSI reduzido de três camadas	16
2.2.2.2.3 Projeto <i>PROWAY</i> (“ <i>Process Data Highway</i> ”)	17
2.2.2.2.4 Projeto MAP (“ <i>Manufacturing Automation Protocol</i> ”)	17
2.2.2.2.5 Barramento de campo (<i>Fieldbus</i>)	19
2.2.2.2.5.1 <i>Profibus</i>	20
2.2.2.2.5.2 <i>Fieldbus Foundation</i>	20
2.3 Tecnologia das Redes <i>Fieldbus</i>	22
2.4 <i>Fieldbus</i> aplicada em controle de processo	25
2.5 Técnicas avançadas (controle adaptativo)	29
2.5.1 Métodos de controle adaptativo	30
2.5.2 Aplicações das técnicas de controle adaptativo com modelo referência	35
2.6 Justificativa	37
3. REDES FIELDBUS	38
3.1 Introdução	38
3.2 Arquitetura do sistema <i>Fieldbus Foundation</i>	40
3.2.1 Modelos de comunicação	40
3.2.2 Topologia de rede	41
3.2.3 Processo de Aplicação (AP)	43
3.2.3.1 <i>Function Blocks Application Process</i>	44
3.2.4 Sistema de gerenciamento <i>Kernel</i>	45
3.2.4.1 Comunicação SMK	46
3.2.4.2 Serviços de SMK	46
3.2.4.2.1 Acesso ao <i>System Management Information Base</i>	46
3.2.4.2.2 Atribuição de endereços e <i>Tag</i>	47
3.2.4.3 Identificação do dispositivo	48
3.2.4.4 Localização remota dos dispositivos e objetos	49
3.2.4.5 Sincronização do <i>clock</i>	49
3.2.4.6 Escalonamento do Bloco Funcional	49
3.2.5 System Management Kernel Protocol Communications (SMKP)	50
3.2.6 Network Management Agent	51
3.2.7 Camada de Aplicação - FMS	51

3.2.8 Subcamada de acesso <i>Fieldbus</i> – FAS	53
3.2.8.1 VCR Cliente/Servidor	54
3.2.8.2 VCR Produtor/Consumidor	54
3.2.8.3 VCR Distribuição de Relatórios	55
3.2.9 Camada de Enlace de dados - DLL	55
3.2.9.1 Controle de acesso ao meio	55
3.2.9.2 Endereço	56
3.2.9.3 Link Active Scheduler - LAS	57
3.2.9.4 Comunicação escalonada	58
3.2.9.5 Comunicação não escalonada	58
3.2.9.6 Manutenção do <i>link</i>	58
3.2.10 Camada Física	59
3.2.11 Camada de Usuário	59
3.2.11.1 Dicionário de Objetos (OD)	59
3.2.11.2 Descrição do Dispositivo - <i>Device Description</i> (DD)	60
 4. TEORIA DE CONTROLE MODERNA	 61
4.1 Introdução	61
4.2 Teoria de controle clássica	62
4.3 Teoria de controle moderno	63
4.3.1 Modelamento por variáveis de estado	63
4.3.2 Lei de controle	66
4.3.2.1 Introdução da entrada de referência com realimentação de estado de ordem plena	68
4.3.3 Observador de estados de ordem plena	70
4.3.4 Projeto do compensador: combinação da lei de controle e observador	73
4.4 Introdução da entrada de referência	75
4.4.1 Estrutura para uma entrada de referência	77
4.4.1.1 Observador Autônomo	78
4.4.1.2 Observador localizador de erro	79
4.4.1.3 Observador com alocação de zeros	80

5. CONTROLE ADAPTATIVO POR MODELO DE REFERÊNCIA	82
5.1 Introdução	82
5.2 Esquemas de controle adaptativos	83
5.2.1 Ganho programado	84
5.2.2 Controle com auto ajuste (<i>Self-Tuning</i>)	85
5.2.3 Controle adaptativo por modelo de referência (CAMR)	86
5.2.3.1 Método do gradiente	87
5.2.3.2 Teoria da estabilidade de <i>Lyapunov</i>	89
5.2.3.2.1 Identificação do problema de controle	90
5.2.3.2.2 O problema de controle	92
5.2.3.2.2.1 Ajuste direto dos parâmetros da planta	93
5.2.3.2.2.2 Controle com realimentação com B_p conhecido	94
5.2.3.2.2.3 Controle com realimentação B_p desconhecido	98
 6. PROJETO DOS NOVOS BLOCOS FUNCIONAIS	 103
6.1 Blocos funcionais	103
6.1.1 Introdução	103
6.1.2 As normas	104
6.1.3 Processo de Aplicação dos Blocos Funcionais	105
6.1.3.1 Introdução	105
6.1.3.2 <i>Resource</i>	106
6.1.3.3 <i>Functions Blocks</i> (Blocos Funcionais)	106
6.1.3.4 <i>Transducer block</i> (Bloco Transdutor - TRD)	106
6.1.3.5 <i>Link Object</i> (objeto Link)	106
6.1.3.6 <i>Resource Block</i> (Bloco Resource – RES)	106
6.1.3.7 <i>View e Trend Objects</i> (Objetos <i>View</i> e <i>trend</i>)	107
6.1.3.8 <i>Alert Objects</i> (Objetos alerta)	107
6.1.3.9 <i>Domain e Program invocation objects</i> (Objetos domínio e programa <i>invocation</i>)	107
6.1.3.10 <i>Action Objects</i> (Objetos de Ação)	107
6.1.3.11 <i>System Time</i> (Tempo do Sistema)	107
6.1.3.12 <i>Function Block Schedule</i> (Escalonamento dos blocos funcionais)	108

6.1.3.13 <i>Network Traffic</i> (Tráfego de Rede)	108
6.1.4 Características dos Blocos Funcionais nos sistemas de controle distribuídos em REDES	108
6.1.4.1 Configuração dos Blocos Funcionais	108
6.1.4.2 Configuração dos <i>links</i> entre os Blocos Funcionais	109
6.1.4.3 Processamento dos alarmes	109
6.1.4.4 Acesso a informações para a interface do operador, diagnóstico e configuração	110
6.1.4.5 Coordenação do funcionamento dos blocos e sua operação	110
6.1.5 Interações Externas	112
6.1.5.1 Interação com Aplicações de Configuração	112
6.1.5.2 Interação com Interfaces Homem-Máquina	113
6.1.5.3 Interação com Outras Aplicações de Configuração	113
6.1.6 Interação com outros Dispositivos de Campo	113
6.1.7 Blocos funcionais no protocolo <i>Fieldbus Foundation</i>	114
6.2 Limitações dos blocos funcionais na aplicação das técnicas de controle	117
6.3 Projeto dos novos blocos funcionais	118
6.3.1 Integração	118
6.3.2 Bloco funcional observador de estados (OE)	120
6.3.2.1 Introdução	120
6.3.2.2 Estrutura e funções do bloco OE	120
6.3.3 Bloco funcional adaptativo por modelo de referência (AMR)	126
6.3.3.1 Introdução	126
6.3.3.2 Estrutura e funções do bloco AMR	126
6.4 Malhas de controle no <i>Fieldbus</i>	131
6.4.1 Malha de controle com realimentação de estado	131
6.4.1.1 Malha de controle por Observação de estados	132
6.4.1.2 Malha de controle com observador autônomo	133
6.4.1.3 Malha de controle com observador autônomo e realimentação de saída	133
6.4.1.4 Malha de controle com alocação de zeros no observador estados	134
6.4.1.5 Malha de controle adaptativa por modelo referência	135

7. TESTES E RESULTADOS	137
7.1 Introdução	137
7.2 Resultados práticos	138
7.2.1 Descrição do sistema físico	138
7.2.2 Descrição do funcionamento do sistema térmico	139
7.2.3 Configuração e supervisão do sistema térmico	139
7.2.4 Modelagem da planta	140
7.2.4.1 Inclusão do tempo morto na função de transferência	145
7.2.5 Modelagem por espaço de estados do sistema térmico	146
7.2.6 Configuração do modelo da planta e dos parâmetros de controle no bloco funcional Observador de estados	146
7.2.6.1 Determinação dos parâmetros da planta	147
7.2.6.2 Determinação do parâmetro K de realimentação	147
7.2.6.3 Determinação do parâmetro L do Observador	148
7.2.7 Estratégia de controle no <i>Fieldbus</i> com o bloco funcional Observador de estados	148
7.2.8 Resultados obtidos	149
7.3 Resultados simulados	152
7.3.1 Estratégia de controle por Observação de estados	153
7.3.2 Resultados obtidos da simulação por Observação de estados	154
7.3.3 Estratégia de controle adaptativa	157
7.3.4 Resultados obtidos da simulação adaptativa	158
8. CONCLUSÃO	162
8.1 Conclusão	162
8.1.1 Trabalhos futuros	164

LISTA DE FIGURAS

FIGURA 1.1 - Sistema de Controle Direto.....	03
FIGURA 1.2 - Sistema Digital de Controle Distribuído.....	04
FIGURA 1.3 - Sistema Digital de Controle Distribuído em comparação ao <i>Fieldbus</i>	05
FIGURA 1.4 - Distribuição do processamento no <i>Fieldbus</i>	05
FIGURA 1.5 - Representação por diagramas de blocos de uma malha controle.....	07
FIGURA 1.6 - Malha de controle com blocos funcionais <i>Fieldbus</i>	07
FIGURA 1.7- Diagrama de blocos inserido na arquitetura <i>Fieldbus</i>	08
FIGURA 2.1 - Modelo de ISO/OSI de sete camadas	16
FIGURA 2.2 - Modelo de ISO/OSI reduzido de três camadas	17
FIGURA 2.3 - Arquitetura MAP/EPA.....	18
FIGURA 2.4 - Arquitetura Mini-MAP	18
FIGURA 2.5 - Estrutura das camadas de rede da <i>Fieldbus Foundation</i>	21
FIGURA 2.6 - Operação das estruturas de rede em um transmissor industrial <i>Foundation</i>	22
FIGURA 2.7 - <i>Fieldbus</i> usada apenas como uma fonte de informações dadas ao algoritmo adaptativo	28
FIGURA 2.8 - Diagrama de otimização da unidade da caldeira.....	29
FIGURA 3.1 - Modelo OSI em comparação com modelo <i>Fieldbus</i>	40
FIGURA 3.2 - Arquitetura dos componentes do sistema <i>Fieldbus Foundation</i>	41
FIGURA 3.3 - Topologia <i>single link</i>	42
FIGURA 3.4 - Topologia <i>multilink</i>	43
FIGURA 3.5 - Estrutura dos Processo de Aplicação	44
FIGURA 3.6 - Arquitetura do sistema de gerenciamento <i>Kernel</i>	46
FIGURA 3.7 - Escalonamento de blocos funcionais	50

FIGURA 4.1 - Diagrama de blocos da equação de estado e de saída.....	65
FIGURA 4.2 - Diagrama de blocos da equação de estado e de saída (matriz D nula).....	66
FIGURA 4.3 - Sistema com a matriz ganho constante na realimentação.....	67
FIGURA 4.4 - Diagrama de blocos de um sistema com realimentação de estados e entrada de referência	69
FIGURA 4.5 - Observador de estados de malha aberta	71
FIGURA 4.6 - Observador de estados de malha fechada	72
FIGURA 4.7 - Mecanismo do controlador e observador.....	75
FIGURA 4.8 - Compensador no caminho de realimentação da malha de controle...	75
FIGURA 4.9 - Compensador no caminho direto da malha de controle.....	76
FIGURA 4.10 - Observador Autônomo.....	79
FIGURA 4.11 - Observador localizador de erro.....	80
FIGURA 4.12 - Observador em alocação de zeros.....	81
FIGURA 5.1- Diagrama de blocos de um sistema com ganho programado.....	84
FIGURA 5.2 - Sistema de controle adaptativo com auto ajuste	85
FIGURA 5.3 - Sistema de controle adaptativo por modelo de referência	87
FIGURA 5.4 - Sistema de controle adaptativo pelo método do gradiente.....	87
FIGURA 5.5 - Sistema de controle adaptativo por modelo de referência com ajuste direto nos parâmetros da planta.....	94
FIGURA 5.6 - Diagrama de blocos de um sistema de controle por modelo de referência com ajuste nos parâmetros do controlador	98
FIGURA 5.7 - Estrutura do controlador com equações de erro não-lineares no ajuste dos parâmetros	99
FIGURA 5.8 - Estrutura do controlador com ajuste de parâmetros usando sinais do sistema	99
FIGURA 5.9 - Diagrama de blocos de um sistema de controle por modelo de referência com ajuste nos parâmetros do controlador com B_p desconhecido	101
FIGURA 6.1 - Componentes do FBAP	105
FIGURA 6.2- Bloco funcional observador de estados.....	122
FIGURA 6.3 – Bloco funcional adaptativo por modelo de referência	128

FIGURA 6.4 – Exemplo de uma malha de controle	131
FIGURA 6.5 – Malha de controle por observação de estados	132
FIGURA 6.6 – Malha de controle com observador autônomo	133
FIGURA 6.7 - Malha de controle com observador autônomo e realimentação de saída	134
FIGURA 6.8 – Malha de controle com alocação de pólos e observador.....	135
FIGURA 6.9 - Malha de controle adaptativa	136
FIGURA 7.1 - Sistema térmico instrumentado com <i>Fieldbus</i>	138
FIGURA 7.2 - Canal <i>Fieldbus</i> no configurador <i>Syscon</i>	139
FIGURA 7.3 - Curva real da resposta para uma entrada degrau de 60%	140
FIGURA 7.4 - Curva de resposta para uma entrada degrau de 60% da curva obtida	141
FIGURA 7.5 - Curva dos dados reais e a curva obtida.....	142
FIGURA 7.6 - Curva de resposta normalizada para uma entrada degrau unitário ..	143
FIGURA 7.7 - Função de Transferência de alimentação direta.....	144
FIGURA 7.8 - Diagrama de blocos da planta e do observador de estado	148
FIGURA 7.9 - Estratégia de controle com o bloco Observador de Estados	149
FIGURA 7.10 - Curvas de resposta do experimento do controle do sistema térmico <i>Setpoint</i> de 70 °C	150
FIGURA 7.11 - Curvas de resposta do experimento do controle do sistema térmico <i>Setpoint</i> de 120 °C	151
FIGURA 7.12 - Curva de resposta do experimento do controle do sistema térmico <i>Setpoint</i> de 170 °C	152
FIGURA 7.13 - Estratégia de controle por observação de estados e a configuração dos parâmetros dos blocos.....	153
FIGURA 7.14 - Curva simulada de resposta no controle do sistema térmico <i>Setpoint</i> de 70 °C	154
FIGURA 7.15 - Curva simulada de resposta no controle do sistema térmico <i>Setpoint</i> de 120 °C	155
FIGURA 7.16 - Curva simulada de resposta no controle do sistema térmico <i>Setpoint</i> de 170 °C	155

FIGURA 7.17 - Curva simulada de resposta no controle do sistema térmico Setpoint de 70 e 140 °C	156
FIGURA 7.18 - Curva simulada de resposta no controle do sistema térmico Setpoint de 100 e 70 °C	156
FIGURA 7.19 - Estratégia de controle adaptativa e a configuração dos parâmetros dos blocos	157
FIGURA 7.20 - Curva de resposta do modelo e da planta para uma entrada degrau de 30%	158
FIGURA 7.21 - Erro para entrada degrau 30% (diferença saída do modelo e da planta)	158
FIGURA 7.22 - Curva de resposta do modelo e da planta para uma entrada degrau de 60%	159
FIGURA 7.23 - Erro para entrada degrau 60% (diferença saída do modelo e da planta)	159
FIGURA 7.24 - Curva de resposta do modelo e da planta para uma entrada degrau de 20, 40 e 60%	160
FIGURA 7.25 - Erro para entrada degrau 20, 40 e 60% (diferença saída do modelo e da planta)	160

LISTA DE TABELAS

TABELA 3.1 - Serviços da FMS	53
TABELA 3.2 - Tamanho máximo dos dados em cada prioridade	56
TABELA 6.1 - Parametrização do bloco Observador de Estados	122
TABELA 6.2 - Parametrização do bloco Adaptativo por Modelo de Referência ...	128

LISTA DE ABREVIATURAS E SIGLAS

A/D	- Analógico/Digital
AI	- <i>Analog Input</i> (entrada analógica)
AO	- <i>Analog Output</i> (saída analógica)
AP	- <i>Application Process</i> (processo de aplicação)
ARTH	- Aritmético
APL	- <i>Application Layer</i> (camada de aplicação)
ASK	- <i>Amplitude-Shift Keying</i> (método por deslocamento de amplitude)
BF	- Blocos Funcionais
CT	- Constante
CS	- <i>Control Selector</i>
CLP	- Controlador lógico Programável
CAMR	- Controle adaptativo por modelo de referência
CAMRI	- Controle adaptativo por modelo de referência indireto
CAMRD	- Controle adaptativo por modelo de referência direto
DO	- <i>Discrete Output</i>
DC	- <i>Device Control</i>
DD	- <i>Device Descriptions</i> (descrições do dispositivo)
DLL	- <i>Data Link Layer</i> (camada de enlace)
DLCEP	- <i>Data Link Connection End Point</i> (ponto final da conexão de link de dados)
DLSAP	- <i>Data Link Connection End Point</i> (ponto de acesso de serviço de link de dados)
DP	- <i>Distributed Control</i> (controle distribuído)

DI	- <i>Discret Input</i>
EE	- Estimador de Estado
EPA	- <i>Enhanced Performace Architecture</i>
FBAP	- Processo de Aplicação dos Blocos Funcionais
FAS	- <i>Fieldbus Access Sublayer</i> (subcamada de acesso ao <i>Fieldbus</i>)
FCS	- <i>Fieldbus systems control</i> (Sistema Controle <i>Fieldbus</i>)
FMS	- <i>Fieldbus Message Specification</i> (camada de aplicação)
FSK	- <i>Frequency-Shift Keying</i> (método por deslocamento de frequência)
FT	- Função de Transferência
ISO	- <i>International Standards Organization</i>
ISP	- <i>Interoperable System Protocol</i>
I/O	- Entrada/Saída
LME	- <i>Layer Management Entity</i> (entidade gerenciadora de comandos)
LAN	- Rede convencional de área local
LAS	- <i>Link Active Scheduler</i> (escalonador ativo de conexões)
LLI	- <i>Lower Layer Interface</i>
LM	- <i>Link Master</i>
MAC	- <i>Medium Access Control</i> (controle de acesso ao meio)
MAU	- <i>Medium Attachment Units</i>
ML	- <i>Manual Loader</i>
MAP	- <i>Manufacturing Automation Protocol</i>
OD	- <i>Object Dictionary</i> (dicionário de objetos)
OSI	- <i>Open System Standards Organization</i>
OUT	- Saída
PA	- <i>Process Automation</i> (processo de automação)
P	- Proporcional
PD	- Proporcional Derivativo
PI	- Proporcional Integral
PID	- Proporcional Integral Derivativo
PDU	- <i>Protocol Data Units</i> (unidades de dados de protocolo)
PSK	- <i>Phase-Shift Keying</i> (método por deslocamento de fase)
PHY	- <i>Physical Layer</i> (camada física)

PT	- <i>Pass Token</i>
SCD	- Sistema Controle Direto
SDCD	- Sistemas Digital de Controle Distribuído
SCF	- Sistema de Controle <i>Fieldbus</i>
SMK	- <i>System Management Kernel</i> (sistema de gerenciamento <i>Kernel</i>)
SMIB	- <i>System Management Information Base</i> (base de dados do gerente do sistema)
SMIK	- <i>System Management Information Protocol</i> (protocolo do núcleo do gerente do sistema)
SISO	- Modelos de sistemas de uma única entrada e uma única saída
MR	- Modelo de Referência
MIMO	- Modelos de sistemas de múltiplas entradas e múltiplas saídas
NMA	- <i>Network management Agent</i> (agente gerenciador de rede)
VFD	- <i>Virtual Field Device</i> (dispositivo de campo virtual)
VCR	- <i>Virtual Communication Relationships</i>

LISTA DE SÍMBOLOS

γ	- Ganho constante
φ	- Derivada do erro em relação aos parâmetros do controlador
θ	Parâmetros do controlador
θ'	- Derivada dos parâmetros do controlador em relação ao tempo
e	- Erro do sistema
$Y(s)$	Sinal de saída entrada no domínio complexo s
$U(s)$	- Sinal de entrada no domínio complexo s
b_n	- Coeficientes do numerador da Função de Transferência
a_n	- Coeficientes do denominador da Função de Transferência
A	- Matriz ($n \times n$), chamada matriz do sistema
B	- Matriz ($n \times r$), chamada matriz de entrada
C	- Matriz ($p \times n$), chamada matriz de saída
D	- Matriz ($p \times r$) que representa o acoplamento direto entre entrada e saída
x	- Vetor dos estados = vetor ($n \times 1$) dos estados de um sistema de ordem n
y	- Vetor de saída = vetor ($p \times 1$) composto pelas saídas definidas
u	- Vetor de entrada = vetor ($r \times 1$) composto pelas funções de entrada
R	- Vetor referência de entrada
K	- Matriz de realimentação de estados
L	- Vetor ganho do estimador de estados
γ_1	- Ganho adaptativo
$\alpha_c(s)$	- Equação característica do sistema
$\alpha_o(s)$	Equação característica do observador

A_m	- Matriz ($n \times n$), chamada matriz de estado do modelo
B_m	- Matriz ($n \times r$), chamada matriz de entrada do modelo
C_m	- Matriz ($p \times n$), chamada matriz de saída do modelo
x_m	Vetor dos estados ($n \times 1$) do modelo
y_m	- Vetor de saída ($p \times 1$) do modelo
A_p	- Matriz ($n \times n$), chamada matriz de estado da planta
B_p	- Matriz ($n \times r$), chamada matriz de entrada da planta
C_p	- Matriz ($p \times n$), chamada matriz de saída da planta
x_p	- Vetor dos estados ($n \times 1$) da planta
y_p	- Vetor de saída ($p \times 1$) da planta
$\hat{x}_p$	- Vetor dos estados ($n \times 1$) da planta observados
ψ	- Vetor erro da matriz A_p entre os valores reais e os observados
ϕ	- Vetor erro da matriz B_p entre os valores reais e os observados
P	- Matriz de estabilidade de Lyapunov

RESUMO

A crescente complexidade do controle dos processos industriais vem exigindo sistemas de controle cada vez mais precisos, confiáveis e versáteis. No sentido de atender a estas exigências, algoritmos diversos de controle e estimação, tais como: técnicas de controle adaptativas, preditivas, estimação paramétrica, filtros de *Kalman*, observação de estados, etc. têm sido desenvolvidas, simuladas e implementadas com relativa facilidade nos modernos sistemas digitais.

Este trabalho propõe uma estratégia de controle em redes *Fieldbus* usando controle adaptativo por modelo de referência através de variáveis de estado, para resolver sistemas complexos. O algoritmo de controle proposto é composto por um observador de estados trabalhando em conjunto com um esquema de controle adaptativo por modelo de referência. As malhas de controle no *Fieldbus* consistem em um conjunto de blocos funcionais padrões existentes, conectados aos novos blocos funcionais propostos e desenvolvidos de acordo com as especificações da norma *Fieldbus Foundation*. Este algoritmo de controle realiza os cálculos de maneira distribuída entre os dispositivos da rede *Fieldbus*, implicando em várias vantagens, tais como: a) perda do supervisor não implica na perda do algoritmo de controle, b) menor tráfego de dados na rede, c) algoritmo de controle que não depende do tempo de processamento do computador.

Os resultados dos testes obtidos são apresentados e demonstraram um alto grau de precisão, destacando-se a estabilidade e aplicabilidade do algoritmo proposto.

Palavras chaves: *Fieldbus Foundation*, blocos funcionais, controle adaptativo por modelo de referência.

ABSTRACT

The rising complexity of the industrial control processes has been claiming more and more accurate, reliable and versatile control systems. Attempting to satisfy this demand, several control and estimation systems algorithms, such as adaptive and predictive techniques, parametric estimation, *Kalman* filtering, state observation, have been designed, simulated and implemented with relative easiness in the modern digital systems.

This work proposes a new control strategy in *Fieldbus* networks, using adaptive control techniques through state variables, to solve complex systems. The proposed control algorithm is based on an states observer concurrently working with a reference model adaptive control scheme. The modified *Fieldbus* network consists of a set of standard function blocks, connected to the proposed new function blocks. These new function blocks comply with the specifications of the *Fieldbus Foundation* norm. This control algorithm accomplishes its calculation in a distributed way among the *Fieldbus* devices. This operating feature results in some advantages, such as: a) any failure in the supervisory system does not imply in the loss of the control algorithm, b) a lower data transmission in the network, c) control algorithm that does not depend on the processing time of the computer.

The obtained results corroborate with the expected advantages of the proposed algorithm, in terms of high degree of accuracy, stability and applicability.

Keywords: Fieldbus Foundation, function blocks, model reference adaptive control.

Capítulo 1

Introdução

1.1 INTRODUÇÃO

No cenário industrial, com exceção de poucas empresas de grande porte, boa parte do parque industrial brasileiro, até a década de 80, empregava técnicas clássicas de controle analógico, em que os ajustes eram muitas vezes realizados de forma precária, sendo baseados apenas no conhecimento do operador e com poucos recursos de tecnologia. Conseqüentemente, isso propiciava aos sistemas de controle um desempenho insatisfatório, comprometendo a eficiência no controle das malhas e contribuindo para a redução da produtividade.

Com o surgimento da grande evolução tecnológica no campo da instrumentação, principalmente com a integração dos microprocessadores, foi possibilitado um avanço nos sistemas de controles industriais, provocando assim, algumas mudanças radicais nas técnicas de projeto.

Novas características foram introduzidas aos sistemas de controle, entre as quais podem ser citadas: o uso de transmissão de dados no formato digital, que além de ser praticamente insuscetível a ruídos, permite a comunicação em redes multiponto, e a possibilidade de realização de algoritmos numéricos de aquisição, filtragem e condicionamento de dados, bem como o de controle e compensação no próprio transmissor.

Na comunicação digital é possível se obter uma rede de comunicação com um

protocolo (organização exigida em toda transferência de dados seriais) que possibilite a transmissão de dados. Esse meio que une a instrumentação (sensor ou atuador) aos equipamentos de controle por um único barramento, no qual todas as informações são transmitidas serialmente e multiplexadas no tempo é chamada *Fieldbus* (FRANCO, 1998).

Atualmente com as redes *Fieldbus*, as aplicações de controle podem variar desde simples controles locais em equipamentos individuais até implementação de malhas de controle, constituídas por instrumentos interligados para um único equipamento, bem como a sistemas digitais distribuídos em estações remotas e interligadas por computadores. Para um determinado setor de uma planta, chegando ao estágio de controle global dos vários setores de uma planta, integrando através de sub-estações e computadores hierarquizados a vários níveis de controle e gerenciamento.

1.2 TECNOLOGIA DE TRANSMISSÃO DE DADOS E REDES DE CAMPO

O controle de processo tem como principal objetivo, manter os processos industriais dentro de seus pontos operacionais mais eficientes, buscar o aumento da produtividade e obter uma melhora na qualidade dos produtos, com diminuição de custos. Para realizar operações de mudanças nas condições do sistema e alcançar tais objetivos previstos, desde a década de 70 até dias atuais, muitas indústrias ainda utilizam o *loop* de corrente 4 a 20 mA.

Com a implementação de controladores e transmissores digitais inteligentes, o padrão de comunicação analógico vem sendo, paulatinamente, substituído por um protocolo digital. Desse modo, baseado nos conceitos de comunicação digital, as redes de campo estão, praticamente substituindo o padrão analógico de 4 – 20 mA.

As redes de campo estão assumindo uma posição de vital importância e destaque, uma vez que, possuem como objetivo o compartilhamento e a interconexão de recursos computacionais, tais como hardware e software, que são geograficamente dispersos em nível local. A arquitetura do sistema de controle passa a se compor por várias estações de controle, aquisição de dados e uma ou mais estações de supervisão interligadas pela rede de comunicação de dados. Essas estações de supervisão se comunicam com as estações de controle e aquisição de dados, manipulando e

gerando conjuntos de informações, que são apresentados aos operadores. Nesse aspecto, a seguir são apresentados os sistemas digitais de controle:

(a) Sistemas de Controle Direto (SCD) - o controle é concentrado em apenas um dispositivo central. A estação central é o elemento essencial à continuidade da operação, além de ser um dispositivo para facilitar e oferecer maiores recursos de interface do operador com o processo. Nos SCD, se o computador parar de funcionar o controle de toda planta é perdido. A Fig. 1.1 mostra um diagrama esquemático do SCD.

FIGURA 1.1 - Sistema de Controle Direto

(b) Sistemas Digitais de Controle Distribuídos (SDCD_s) – o controle pode ser distribuído em diversas placas eletrônicas em uma rede. Nesses sistemas, vários controladores são conectados a uma rede digital até estações supervisórias e de configuração. Geralmente utilizam o tradicional sistema de transmissão de dados por *loop* de corrente em 4 a 20 mA, para acionarem transmissores e atuadores. Embora os sistemas SDCC_s sejam capazes de processar sinais de modo digital, a transmissão analógica faz com que esses sistemas fiquem limitados à linhas de dados, dedicadas ponto-a-ponto entre controladores e transmissores, isto é, cada transmissor necessita de um par de fios para se conectar ao controlador. Como consequência, existe a necessidade de serem utilizados módulos de I/O (Entrada/Saída) entre controladores e transmissores, elevando o custo e o grau de complexidade de instalação,

configuração e manutenção. Os algoritmos de controle nos SDCD_s são executados exclusivamente nos módulos controladores, cabendo aos transmissores apenas o condicionamento dos sinais. Desse modo, uma malha de controle é processada em um único controlador. No caso de um módulo controlador apresentar defeito e não for redundante será perdida apenas uma parte do controle da planta. A Fig. 1.2 mostra um diagrama do SDCD_s.

FIGURA 1.2 - Sistema Digital de Controle Distribuído

(c) Sistemas de Controle *Fieldbus* (SCF) - o controle é verdadeiramente distribuído nos dispositivos de campo. Com a evolução dos componentes eletrônicos, foi possível ser adicionada maiores funcionalidades aos transmissores devido ao aumento da capacidade das CPUs de memórias e pela constante evolução nas técnicas de encapsulamento de circuitos integrados. Essa evolução, permitiu a implementação de protocolos de comunicação digital nos transmissores e ofereceu condições de garantia maiores para funções em termos de aquisição, linearização e compensação das variáveis.

Em uma rede *Fieldbus*, todos os dados são processados e transmitidos de forma digital, possibilitando assim, que os transmissores sejam conectados uns aos outros e com estações supervisórias e configuradoras, sendo utilizadas várias topologias físicas de rede, em modo multiponto. A Fig. 1.3 mostra um esquema de comparação entre o Sistema de Controle Digital Distribuído e o Sistema de Controle *Fieldbus*.

FIGURA 1.3 - Sistema Digital de Controle Distribuído em comparação ao *Fieldbus*

Nas redes *Fieldbus* atuais é observada a migração dos algoritmos de controle de malhas dos controladores para os transmissores, em que pela primeira vez, as malhas passam a ser executadas no campo e distribuídas entre os vários transmissores, de forma verdadeira. A Fig. 1.4 mostra a distribuição de processamento no *Fieldbus*.

FIGURA 1.4 - Distribuição do processamento no *Fieldbus*

De fato, em sistemas *Fieldbus*, além das tarefas de controle em si, também são realizados todos os serviços de monitoramento e supervisão da planta e de configuração dos transmissores nas malhas, com grande quantidade de informações adicionais sobre o processo, incluindo qualidade e *status* dos dados do processo, por exemplo. É permitido incluir entre as informações adicionais diversas delas relativas, a qualidade de operação dos dispositivos de campo, proporcionando a implementação de manutenção preditiva, a partir de dados quantitativos de desempenho e qualidade de operação dos dispositivos monitorados pela rede. Com isso é possível perceber quando um dispositivo começa a apresentar tendência de falha, podendo assim programar-se a sua substituição de forma adequada. Essa possibilidade aumentou significativamente a segurança na operação das plantas controladas por meio da tecnologia *Fieldbus*, bem como o aumento da produtividade, em níveis até então não alcançados, através da diminuição das quebras não previstas. Além deste ganho, obtido pelo aumento do conhecimento do comportamento dos equipamentos de campo, quando monitorados via rede *Fieldbus*, torna-se fundamental observarmos, que é apresentada também a possibilidade da diminuição das redundâncias de equipamentos, por necessidade de segurança das plantas.

1.3 MALHAS DE CONTROLE COM A TECNOLOGIA *FIELDBUS* ATUAL

As estratégias de controles de processo via redes *Fieldbus* são realizadas através de blocos funcionais, que interligam os dispositivos de campo e processam os sinais diretamente nos nós da rede. Os blocos funcionais são agrupamentos de funções lógicas de controle inter-relacionadas e identificados através de seus parâmetros de entrada e saída.

Existe, atualmente, mais de vinte blocos funcionais padronizados para execução de estratégias de controle, entre as quais podemos citar o controle em cascata, relação, limites cruzados, etc.. Essas estratégias de controle são configuradas através da ligação dos blocos funcionais entre os vários dispositivos de campo do sistema. A Fig 1.5 mostra a representação por diagramas de blocos de um sistema de controle de malha fechada de uma planta compensada por um controlador tipo PID. A mesma configuração desse sistema de controle representado na tecnologia *Fieldbus* através da ligação dos blocos funcionais pode ser visto pela Fig. 1.6.

FIGURA 1.5 - Representação por diagramas de blocos de uma malha controle

FIGURA 1.6 - Malha de controle com blocos funcionais *Fieldbus*

Nessa representação, os blocos funcionais representados por AI (*Analog Input*), PID (*Proportional Integral Derivativo*) e AO (*Analog Output*) realizam as funções de aquisição de variável analógica, compensação PID e geração de sinal de atuação analógico, respectivamente. Cada bloco possui entradas e saídas, que são transmitidas de forma digital via rede ou de modo local. Assim, é possível transmitir via rede diferentes tipos de variáveis.

É possível perceber que os blocos funcionais estão inseridos nos dois transmissores, assim a malha de controle é computada diretamente no campo, de modo em que não é mais necessário, nesse caso, um dispositivo controlador, como pode ser observado na Fig. 1.7.

A Fig. 1.7 mostra também a inserção da malha de controle por diagrama de blocos nos blocos funcionais *Foundation Fieldbus* integrados aos transmissores de campo em rede.

FIGURA 1.7- Diagrama de blocos inserido na arquitetura *Fieldbus*

1.4 A IMPORTÂNCIA DE ALGORITMOS AVANÇADOS DE CONTROLE

Como pôde ser visto anteriormente, apesar de todo avanço tecnológico, as malhas de controle do *Fieldbus* ainda estão sendo realizadas através dos compensadores clássicos, como por exemplo, o controlador Proporcional (P), Proporcional Integral (PI), Proporcional Derivativo (PD) e Proporcional Integral Derivativo (PID). A razão pela qual isso acontece, se dá, pela inexistência de ferramentas (blocos funcionais) que podem realizar os processamentos desejados de técnicas mais avançadas, DOMINGUES (2002).

Nesse aspecto, o desenvolvimento de novas estratégias de controle avançado para o *Fieldbus*, utilizando os algoritmos de controle e estimação existentes para uso integrado e versátil, na forma de objetos (com propriedades e serviços específicos) em um ambiente de tempo real e aplicados a processos industriais, pode nos levar a resultados muito interessantes. Aplicar técnicas avançadas de controle, utilizando a tecnologia *Fieldbus* representa muito mais do que apenas uma solução alternativa aos sistemas de controles atuais. Optando por essa nova solução, várias vantagens podem ser obtidas, na quais algumas delas são citadas abaixo:

a) Sistema de controle mais eficiente, devido a maior flexibilidade e versatilidade do sistema *Fieldbus* que, possibilita a implantação de algoritmos mais elaborados, distribuídos, processando em paralelo e com malhas mais elaboradas, que permitem a especificação de novas funções de controle extremamente difíceis de serem implementadas nos SDCD;

b) Introdução de soluções pouco usadas até hoje em controle de processo,

dessa forma, como controle adaptativo, que poderão melhorar o desempenho do processo em várias situações;

c) Controle com mais de uma variável simultaneamente, não havendo a necessidade de realizar o desacoplamento das variáveis do processo;

d) Integração do sistema de controle, monitoração, segurança, manutenção e supervisão da planta, possibilitando novas soluções, até então, não disponíveis nos SDCD;

A proposta de uma nova maneira de se realizar controle por meio de técnicas avançadas, com processamento distribuído em redes *Fieldbus*, cria uma nova classe de soluções para os problemas até então não estudados. A solução de processamento distribuído, com a possibilidade de processamento paralelo e conjugados com a necessidade de se estabelecer os “*scadules*” (escalonamento temporal do processamento dos blocos funcionais), sendo utilizados os mesmos dispositivos de campo atualmente usados, configura-se como arquiteturas de hardware conjugadas, com soluções *de software* ainda não estudadas. Nesse contexto, o tema a ser estudado vem contribuir com novas soluções nesse campo.

1.5 OBJETIVO

Este trabalho tem como objetivo, elaborar uma estratégia avançada de controle e supervisão adaptativa em sistemas por variáveis de estado, aplicada às redes *Fieldbus*, para que possa atender às exigências mais complexas de uma forma flexível e versátil, tanto em relação às suas estruturas globais quanto em termos de ferramentas disponíveis. Essa estratégia adaptativa com observador de estados será composta por novos blocos funcionais desenvolvidos, simulada mediante o *software Matlab* e testada no controle de uma planta térmica. Nos testes realizados serão consideradas as mesmas situações de uma planta real, em que os cálculos serão distribuídos entre os dispositivos da rede *Fieldbus*, ao contrário da tecnologia atual, quando o processamento de algoritmos avançado de controle é executado no supervisor do computador central. Os resultados a serem apresentados evidenciarão a aplicabilidade e estabilidade da estratégia, bem como suas vantagens e desvantagens.

1.6 ORGANIZAÇÃO DO TRABALHO

No capítulo 2 é apresentado o estado da arte referente à evolução tecnológica em controle de processo e os trabalhos desenvolvidos mais recentemente. Também serão focalizados os trabalhos mais atuais, sobre as redes de comunicação de dados, em especial o *Fieldbus*, aplicados a área de controle de processos.

No capítulo 3 são descritos os principais componentes da arquitetura do sistema *Fieldbus Foundation*, bem com as camadas de comunicação ressaltando o funcionamento e as características encontradas em cada um delas.

No capítulo 4 é abordada a representação por espaço de estado e algumas técnicas empregadas na observação de estados de sistemas multivariáveis.

No capítulo 5 são descritos vários esquemas de controle adaptativos, bem como o método que será empregado nesse trabalho como bloco funcional.

No capítulo 6 é desenvolvido o projeto dos novos blocos funcionais, de acordo com a norma *Fieldbus Foundation*.

No capítulo 7 são mostrados os resultados obtidos pelas novas estratégias de controle, para uma variedade de situações. Além disso, são feitas algumas considerações sobre a eficácia do método.

No capítulo 8 são relatadas as conclusões do trabalho, destacando-se as vantagens e desvantagens da estratégia abordada.

Capítulo 2

Estado da Arte – Controle de Processos

Neste capítulo apresenta-se a evolução tecnológica em controle de processo e os trabalhos desenvolvidos mais recentemente.

2.1 INTRODUÇÃO

O controle de processo é utilizado basicamente para manter constante, em torno de um valor desejado, as variáveis importantes para a boa operação de uma planta industrial. O conceito de manter uma variável constante está ligado na prática a reduzir a variabilidade, ou seja, seu desvio do ponto desejado, VALDMAN (1998).

Para permitir que o controle reduza a variabilidade de uma variável de interesse, fazendo ajustes em atuadores, como válvulas de controle e variadores de velocidade novas tecnologias vem sendo desenvolvidas no decorrer dos anos.

Esse capítulo tem como meta relatar a evolução científica dessas novas tecnologias, enfatizando desde a transmissão analógica de sinais, até os dias de hoje, em que a transmissão é digital e as plantas são controladas por redes de comunicação utilizando técnicas de controle avançadas. Também serão focalizados os trabalhos mais recentes sobre as redes de comunicação de dados, em especial o *Fieldbus*, aplicados a área de controle de processos.

2.2 EVOLUÇÃO TECNOLÓGICA EM CONTROLE DE PROCESSO

2.2.1 TRANSMISSÃO DE SINAIS

A transmissão de sinais é um aspecto fundamental para realizar o controle das grandezas físicas envolvidas no processo, devido principalmente aos componentes dos modernos sistemas de controle digital (sensores, controladores e atuadores) estarem normalmente separados entre si e localizados em lugares distantes. Um meio de transmissão confiável, representa uma maior segurança com relação a precisão das medidas realizadas e conseqüentemente a qualidade da malha de controle que não é afetada.

No início, a transmissão era realizada de maneira analógica, em que o sinal variava continuamente dentro de uma faixa pré-estabelecida, com a intensidade do mesmo, representando o valor numérico da variável transmitida no processo. O primeiro padrão analógico surgido foi o pneumático, que varia na faixa de 3 a 15 Psi. Vários problemas foram encontrados nesse tipo de comunicação: relação da constante de tempo da transmissão do sinal mediante a do processo controlado, e a baixa imunidade do sinal para variações do ambiente.

Na década de 60, surgiram os transmissores eletrônicos, que geravam sinais elétricos (tensão, frequência e corrente) para representar o valor da grandeza medida. Problemas de atenuação do sinal na linha e de suscetibilidade a ruídos eletromagnéticos foram encontrados na transmissão dos sinais por tensão e frequência respectivamente. A transmissão dos sinais por corrente foi o método mais difundido e o que apresentou melhor desempenho. No entanto, em 1975 através da norma SP-50, a ISA adotou a faixa de 4 a 20 mA como padrão industrial. Devido às vantagens apresentadas nesse sistema, nos dias atuais, ainda existe um grande número de indústrias que utilizam a malha de 4 a 20 mA para realizar o controle.

Na década de 70, com o desenvolvimento dos microprocessadores, a transmissão digital começou a ser utilizada. A forma mais simples e mais usada é a codificação binária, em que os sinais transmitidos podem assumir somente dois valores distintos, conhecidos como “0” e “1”.

As três formas básicas de se transmitir a informação binária são: método por amplitude (*Amplitude-Shift Keying*, ASK), por deslocamento de frequência

(*Frequency-Shift Keying*, FSK) e por deslocamento de fase (*Phase-Shift Keying*, PSK). Códigos multi níveis também podem ser utilizados para a transmissão de informações, em que o sinal poderá assumir três ou mais valores discretos. A razão para a utilização de sinais digitais multi níveis é a aquisição de tempo na transmissão e a economia da largura de banda da linha de transmissão. Em contra partida, é exigida uma difícil decodificação e apresentada uma maior susceptibilidade à ruídos.

2.2.2 SISTEMAS DE TRANSMISSÃO DE SINAIS

Até alguns anos atrás, os sistemas de transmissão de sinais eram munidos de vários amplificadores, espalhados por toda a malha de controle, que possibilitavam a chegada de sinais até um controlador central, em que o mesmo, realizava as correções necessárias nos sinais.

A partir do início da década de 80 começaram a surgir os primeiros transmissores dotados de microprocessadores, conhecidos como transmissores inteligentes (*smart transmitters*).

2.2.2.1 TRANSMISSORES INTELIGENTES

Um típico transmissor desse tipo possui: unidade de processamento, interface de comunicação, transdutor(es) e unidade de tratamento de sinais. Nos transmissores inteligentes, o sinal é adquirido pelo transdutor, transformado por um conversor A/D, processado o valor digital e posteriormente, transmitido diretamente na forma digital, ou então alterado por um conversor D/A e enviado na forma analógica. A evolução dos transmissores inteligentes se dá, entre outros fatores, com o aumento da capacidade de memória e de processamento interno. Essa evolução permitiu a classificação dos mesmos, em duas gerações:

A primeira geração fazia a conversão A/D do sinal vindo do sensor, entregava ao microprocessador para tratamento e então reconvertia o sinal tratado para a forma analógica (D/A). Visando assim, manter a compatibilidade com os transmissores tradicionais dos sistemas de controle por *loop* de corrente de 4 a 20 mA. A comunicação com o microprocessador teve como característica a sobreposição do sinal de comunicação digital. Apesar de possuírem algumas vantagens do ambiente

digital, como aumento da precisão e menor susceptibilidade à variação de temperatura de trabalho, os transmissores da primeira geração eram essencialmente analógicos, devido a transmissão receber essa mesma característica.

A segunda e atual geração de transmissores inteligentes se caracterizava pela eliminação definitiva da transmissão analógica, tornando-se puramente digital. Com a eliminação da grande maioria dos componentes analógicos, esses transmissores possuem todas as vantagens do ambiente e da transmissão digital citadas anteriormente. A precisão da malha de controle foi aumentada com a eliminação das conversões D/A e A/D, necessárias para a transmissão de dados. Outra vantagem importante foi observada como consequência da gradativa migração de parte do processamento de sinais para os transmissores, aliada com a comunicação digital bidirecional transmissor-controlador: a distribuição das funções de controle, gerando assim, um aumento na capacidade e no desempenho dos controladores e dos SDCDs.

2.2.2.2 PROTOCOLOS E PADRONIZAÇÃO

Com o grande avanço proporcionado pelos transmissores inteligentes, no campo de controle de processo e a crescente utilização destes equipamentos, surgiu um problema, no que diz respeito a diferenças entre aqueles vindos dos vários fabricantes. Nesse aspecto, uma planta de controle fica restrita ao fabricante que forneceu os equipamentos, não podendo integrar modelos novos, caso esses não sejam do mesmo fabricante.

Essa situação desfavorável aos usuários e fabricantes, aliada ao contínuo avanço dos microprocessadores, que tornava os transmissores industriais cada vez mais compactos e com mais recursos, motivou grupos de empresas e usuários a buscarem a criação de padrões e normas para o projeto de sistemas de comunicação digital.

A partir dessa necessidade, indústrias e organismos de normalização nacionais e internacionais, iniciaram o desenvolvimento de um novo conceito em redes de comunicação digital. Essas pesquisas criaram as bases do que é conhecido atualmente por *Fieldbus*.

2.2.2.2.1 MODELO DE REFERÊNCIA ISO/OSI

O modelo de referência ISO/OSI (*International Standards Organization / Open System Interconnection*) de sete camadas de rede é atualmente uma das bases para o desenvolvimento de redes de computadores. O modelo é basicamente constituído de um conjunto de sete camadas independentes e hierarquicamente sobrepostas, em que cada camada é responsável por uma classe de serviços e se comunica com as camadas vizinhas, através da prestação desses serviços. As sete camadas, em ordem crescente de hierarquia são:

- 1) Camada Física (*Physical Layer*);
- 2) Camada de Enlace (*Data Link Layer*);
- 3) Camada de Rede (*Network Layer*);
- 4) Camada de Transporte (*Transport Layer*);
- 5) Camada de Sessão (*Session Layer*);
- 6) Camada de Apresentação (*Presentation Layer*);
- 7) Camada de Aplicação (*Application Layer*).

Os princípios aplicados para chegar às sete camadas mencionadas acima de acordo com TANENBAUM (1996) foram os seguintes:

- (a) Uma camada deveria ser criada onde um nível diferente de abstração é necessário;
- (b) Cada camada deveria realizar uma função bem definida;
- (c) A função de cada camada deveria ser escolhida com o intuito de definir um protocolo padronizado internacionalmente;
- (d) Os limites das camadas deveriam ser escolhidos para minimizar o fluxo de informação através das interfaces;
- (e) O número de camadas deveria ser suficientemente grande para que funções distintas não fossem passadas juntas para a mesma camada sem necessidade, e pequena o bastante para que a arquitetura não viesse a ser inconveniente.

A Fig. 2.1 representa o mecanismo de comunicação vertical entre as camadas vizinhas de um mesmo ponto de rede e o mecanismo de comunicação horizontal entre as camadas semelhantes de dois pontos de rede distintos.

FIGURA 2.1 - Modelo de ISO/OSI de sete camadas

2.2.2.2.2 MODELO ISO/OSI REDUZIDO DE TRÊS CAMADAS

O modelo reduzido ISO/OSI serviu de referência para o desenvolvimento de diferentes protocolos de redes industriais *Fieldbus*. Para se adaptar as necessidades de um *Fieldbus*, a estrutura de sete camadas foi limitada a somente três camadas: a camada física, a camada de enlace e a camada de aplicação, como demonstrado na Fig. 2.2.

As camadas intermediárias (camada de rede, transporte, sessão e apresentação) foram eliminadas devido aos serviços prestados por essas camadas não serem necessários em um *Fieldbus* ou em qualquer outra rede local. A exclusão dessas camadas torna também o processo de comunicação mais rápido.

FIGURA 2.2 - Modelo de ISO/OSI reduzido de três camadas

2.2.2.2.3 PROJETO PROWAY (“PROCESS DATA HIGHWAY”)

O desenvolvimento dos protocolos para ambientes de chão de fábrica começou em meados dos anos 70. Em 1976 o grupo de estudos IEC TC 65C/WG6, em parceria com a ISA SP-72, iniciou os trabalhos de um projeto denominado PROWAY. Esses trabalhos resultaram nos protocolos PROWAY-A e PROWAY-B. Em 1980 a IEEE criou o projeto 802 e futuramente, em 1984, em parceria com membros da ISA, foi definido o IEEE 802.4 e o ISA SP-72.01 em 1985. Esse último ficou conhecido como PROWAY-C. O projeto PROWAY apresentou algumas deficiências graves relacionadas a tempos de resposta e confiabilidade, porém foi de grande importância, pois lançaram as bases para o desenvolvimento de futuros projetos.

2.2.2.2.4 PROJETO MAP (“MANUFACTURING AUTOMATION PROTOCOL”)

A arquitetura de comunicação MAP (*Manufacturing Automation Protocol*) foi originalmente proposta pela General Motors no início dos anos 80 como padrão de comunicação no ambiente industrial. Essa arquitetura de sete camadas consistindo um modelo OSI resultou em uma rede com alto grau de compatibilidade e uma faixa abrangente de aplicações, porém com tempo de resposta relativamente elevado para

aplicações de controle de processo, devido ao grande processamento envolvido. Esse fator contribuiu para a proposta de novas arquiteturas, a MAP/EPA e a mini-MAP.

A configuração MAP/EPA (*MAP/Enhanced Performance Architecture*) consiste em um nó MAP com melhor desempenho em termos de resposta, assim como demonstrada na Fig. 2.3.

FIGURA 2.3 - Arquitetura MAP/EPA

A configuração mini-MAP implementa um conjunto de protocolos reduzidos, capaz de prover comunicações no nível de controle direto, com tempo de resposta e segurança exigidos pela aplicação a um custo compatível. A arquitetura Mini-MAP contém somente três camadas (física, enlace e aplicação) da estrutura MAP/EPA e pode ser vista através da Fig. 2.4.

FIGURA 2.4 - Arquitetura Mini-MAP

2.2.2.2.5 BARRAMENTO DE CAMPO (*FIELDBUS*)

O crescente aumento da automação industrial e a necessidade de sistemas com tempo de resposta pequenos, principalmente nas aplicações de controle em tempo real, deixaram claras as deficiências das redes locais quando usadas com essas finalidades. A solução foi então, o desenvolvimento de um o Barramento de Campo ou *Fieldbus* como é conhecido, especialmente projetado para aplicação no chão de fábrica, com objetivo de realizar a comunicação entre os dispositivos de campo e os sistemas de controle, PACHECO (1998).

O *Fieldbus* é definido como um elo de comunicação serial digital entre os dispositivos de automação primários, localizados juntos ao processo e o nível de controle imediatamente superior na sala de controle, com maior capacidade de processamento. O *Fieldbus* já foi concebido para o nível de controladores e transmissores.

Organismos de padronização pioneiros nos trabalhos com *Fieldbus*, como o IEEE, o IEC e a ISA, iniciaram os trabalhos por elaborar um conjunto de requisitos iniciais que o novo protocolo deveria suportar, no que se diz respeito a tempos de resposta, topologia física, redundância, confiabilidade, número de dispositivos por ramo de rede, método de acesso ao meio físico, alimentação dos transmissores, comprimento do barramento e métodos de configuração.

A padronização internacional do barramento, de campo tornará fácil a elaboração de interfaces para os diversos tipos de aplicações industriais, baseados em um mesmo protocolo, além de tornar o mercado de sistemas aberto, ou seja, garantir a interoperabilidade de equipamentos de diferentes fabricantes.

Esse objetivo, porém, é difícil de ser alcançado, pois os processos industriais são muito diversos, além do que estão também envolvidos muitos interesses econômicos da parte de fabricantes de instrumentos e de sistemas.

No cenário atual, entretanto, algumas normas se destacam e possuem grande aceitação no mercado. A nível de sistemas para processo industriais contínuos, os protocolos de maior destaque e bases instaladas atualmente são: *Profibus (Process for Fieldbus)* alemão e o *Fieldbus Foundation*. A seguir serão apresentadas algumas características destes protocolos.

2.2.2.2.5.1 PROFIBUS

O protocolo *Profibus* foi desenvolvido à partir de 1987, na Alemanha, por um conjunto de quinze empresas e cinco institutos de pesquisa, financiados pelo governo alemão, por iniciativa das indústrias *Bosch*, *Klockner-Moeller* e *Siemens*. Como resultado desses trabalhos, foi publicada em 1991 a norma DIN 19254.

As especificações do *Profibus* são baseadas no consagrado modelo reduzido ISO/OSI de três camadas e nas normas internacionais IEC-955, para o gerenciamento das camadas, IEEE 802.2 para a LLI (*Lower Layer Interface*) e o ISO IS 9506 para a camada de aplicação FMS (*Fieldbus Message Specification*).

Desde 1996 o *Profibus* é adotado como parte da norma europeia EN 50 170 e atende a uma vasta gama de aplicação, pois o protocolo é composto por três versões totalmente compatíveis: o *Profibus-DP* (*Distributed Control*), otimizado para comunicação de alta velocidade e especialmente projetado para sistemas de controle discreto e de I/O distribuído, sendo que, das três versões é a mais difundida; o *Profibus-FMS* (*Fieldbus Message Specification*), dedicado a comunicação no nível de células de manufatura para tanto, tem estrutura mais complexa e possui serviços mais elaborados e não determinísticos; e o *Profibus-PA* (*Process Automation*) projetado para controle de processos contínuos e adaptado à áreas classificadas.

2.2.2.2.5.2 FIELDBUS FOUNDATION

O protocolo *Fieldbus Foundation* foi proposto inicialmente em 1994, por uma fundação internacional de mesmo nome, sediada nos Estados Unidos e composta pela união de duas organizações de *Fieldbus* existentes : a *ISP* – (*Interoperable System Protocol*) e uma divisão da *WorldFIP* francesa e mais um grupo de oitenta e cinco empresas.

Aliados aos organismos internacionais de normalização IEC e ISA, a fundação propôs uma arquitetura também baseada no modelo ISO/OSI reduzido de três camadas. O protocolo da camada física é o proposto atualmente na norma IEC 61158-2. A camada de Enlace baseia-se em um subconjunto da norma IEC FDIS

61158-3 e 4 é a camada de Aplicação na norma IEC FDIS 61158-5 e 6.

Existe ainda definida no *Fieldbus Foundation*, uma camada superior a camada de Aplicação, denominada camada de Usuário: a *User Layer*. Nessa camada estão definidas estruturas que realizam funções do controle e supervisão da planta em si, configuradas em uma arquitetura que muito se assemelha àquela encontrada no *Profibus-PA*.

Baseada nas normas IEC 65C e ISO 1131-3, a camada de usuário do *Fieldbus Foundation* possui entre suas estruturas, algumas derivadas do *Profibus* e do *Hart*.

O protocolo *Fieldbus Foundation*, ao contrário do *Profibus*, não possui versões diferentes e foi concebido para a aplicação em controle de processos contínuos. A estrutura de suas camadas está representada pela Fig. 2.5.

FIGURA 2.5 - Estrutura das camadas de rede da *Fieldbus Foundation*

Essas camadas podem ser representadas com enfoque na estrutura de operação no transmissor, de um ponto de vista computacional como na Fig. 2.6.

FIGURA 2.6 - Operação das estruturas de rede em um transmissor industrial

Foundation

As especificações *Foundation* ainda não estão finalizadas, no entanto, fabricantes e usuários atualmente trabalham para definir e aprovar os capítulos inacabados da norma. Por outro lado, é natural que mesmo as especificações que já foram aprovadas passem por um contínuo processo de evolução, observa-se que periodicamente novas revisões e novos capítulos são publicados pela *Foundation*, na medida em que os requisitos dos sistemas de automação vão se tornando cada vez mais complexos e que os fabricante e usuários vão acumulando experiência na operação e instalação de sistemas.

Uma descrição das estruturas que compõe o protocolo será apresentada no capítulo 3, com o propósito de se entender melhor o funcionamento geral do sistema.

2.3 TECNOLOGIA DAS REDES *FIELD BUS*

No decorrer dos recentes anos, o campo da instrumentação e do controle de processo em tempo real, tem estado em uma direção distante do sistema de controle centralizado. O controle distribuído, usando a tecnologia *Fieldbus* é só o primeiro passo para a aplicação de dispositivos inteligentes de campo. Nesse aspecto, esse assunto tem sido de grande importância nos últimos anos e de interesse de muitos pesquisadores e especialistas na área, PACHECO et al (2002).

ZHAO (1996) evoluiu o método de controle *Fuzzy* para Sistema Controle *Fieldbus* (FCS), através de redes de neurais. Ele relata que a precisão do FCS não só depende dos resultados da lógica, que são resolvidos pelo sistema, como também do tempo determinístico dos resultados.

Nos métodos, são utilizadas as regras *Fuzzy*, bem como dados numéricos.

A arquitetura proposta da rede neural traçou um vetor de entrada *Fuzzy* para uma saída *Fuzzy*. Um algoritmo de aprendizagem foi deduzido da saída atual *Fuzzy* e da saída designada *Fuzzy*, e os resultados dos estudos solucionaram o tipo de problema de controle em FCS, devido aos dispositivos de campo não só precisarem do conhecimento de especialistas, mas também das experiências dos processos atuais.

MAHALIK & MOORE (1997) comparam os sistemas centralizados com os distribuídos, diferenciando uma rede convencional de área local (LAN), de uma rede *Fieldbus*. Eles descrevem alguns padrões *Fieldbus*, destacando as vantagens de usá-lo no controle de processos em tempo real e discutem quais pontos devem ser considerados antes de escolher uma rede *Fieldbus*.

Apresentam, além disso, uma metodologia do desenvolvimento baseada no projeto *Fieldbus*, usada para construir um sistema de controle industrial distribuído em tempo real, bem como a apresentação de um caso, em que é aplicada a tecnologia *Lonworks*.

O sistema *Fieldbus* possibilita funções de aquisição de dados e controle de malha fechada, para serem executadas diretamente nos instrumentos *Fieldbus*. Com *Fieldbus*, muitas das funções contidas nos sistemas de controle tradicionais, tais como: blocos funcionais, escalonamento determinístico, banco de dados distribuído e alarmes, podem migrar para os dispositivos e para a própria rede.

O *Fieldbus* provê as bases para uma nova classe de sistemas de controle em pequena escala, em que a capacidade do controle reside nos dispositivos.

GLANZER & SANTORI (1998) relatam funções como: configuração do sistema, manutenção, diagnósticos e interface homem/máquina, que podem ser executadas através de sistemas baseados em PC.

A tecnologia da *Fieldbus Foundation* é uma especificação aberta para sensores, atuadores e analisadores. Isso inclui definições que permitem ao controle ser residente nos dispositivos de campo. Além do mais, aceita que muitas funções não controladas passem a ser residentes, nos dispositivos inteligentes, devido ao acesso de informações ser agora padronizado.

HODSON, W, R. (1998) enfoca suas atenções nas funções não controladas, incluindo diagnósticos, operação de processo e funções de manutenção que o *Fieldbus* consente ser residente no campo e em outros dispositivos localizados remotamente.

DRWAL & WEREWKA (1999) relatam um modelo desenvolvido, que descreve o fluxo de informação de estação para estação e vice-versa. Eles consideram a influência de restrições no projeto de um sistema de controle distribuído em tempo real, no qual depende de muitos fatores, sendo que um deles é a estrutura da rede.

A influência da estrutura da rede sob o comportamento em tempo real foi analisada e alguns exemplos a ilustram por se encontrar em condições nesse tempo. O problema da otimização para determinar a estrutura mais adequada foi também investigado.

A função da comunicação de campo é um dos elementos do sistema de informação de uma empresa. A informação gerada em dispositivos de campo é usada, principalmente para controle e administração de manutenção.

Yokogawa atualmente provê dispositivos de comunicação de campo híbridos, baseado no protocolo BRAIN para esses propósitos.

MORIOKA (1999) descreve o conceito de projeto para produtos de comunicação digital de campo fundamentado no *Foundation/sup TM/ Fieldbus*.

ENGELMANN et. al, (2001) apresentaram uma possível solução para resolver problemas de escalonamento de mensagens cíclicas e acíclicas no sistema *Fieldbus* com algoritmo genético. O algoritmo realiza o escalonamento em função do processamento dos blocos funcionais, determinando o melhor instante para transmissão das mensagens.

2.4 *FIELD*BUS APLICADA EM CONTROLE DE PROCESSO

O *Fieldbus* foi projetado especificamente para atender às necessidades da indústria de processo, incluindo variáveis de processo, controle de processo determinístico em tempo real e diagnósticos. Aplicações em refinarias, destilarias, caldeiras, agricultura, indústrias químicas, farmacêuticas e alimentícias são apenas alguns exemplos.

O uso de novos sistemas de controle automáticos em aplicações automotivas, adicionou um vasto número de dispositivos mais simples, conduzindo a alguns problemas relativos à complexidade na instalação elétrica dos veículos. Para resolver esses problemas, uma necessidade comum foi percebida pelo sistema alternativo de interconexão.

CAVALIERI et al. (1996) realizaram o estudo de um caso da aplicação do *IEC/ISA Fieldbus*, o qual é um sistema particular de comunicação serial digital, aos sistemas de controle presente no carro. Os autores tratam com o problema da administração do controle do tráfego, explorando os aspectos *IEC/ISA Fieldbus* e avaliando seu comportamento em diferentes condições de tráfego. Apesar da aplicação particular considerada, os resultados obtidos têm uma validade geral e mostram a habilidade do *Fieldbus* para satisfazer as exigências de comunicação dos sistemas de tempo críticos.

Com várias aplicações em uma vasta área industrial, o *Fieldbus* está provando ser uma solução confiável para o controle de processo. DULTRA & VELONI (1996) encontraram uma ótima solução para controlar e operar caldeiras em uma usina de açúcar, através de uma combinação entre a tecnologia *Fieldbus*, PLC e o *Software MMI*. O uso de *Fieldbus* permitiu a distribuição das funções de controle entre os dispositivos de campo com consideráveis economias no hardware, cablagem e instalação. O *Fieldbus* foi configurado com um software de interface gráfica interativa, que reduziu o tempo de configuração e não exigiu nenhum treinamento especial.

FRANCO et al. (1999) apresentam uma alternativa para a substituição do cabeamento elétrico convencional de um automóvel por uma rede digital. O projeto utiliza módulos de entrada e saída conectados a um barramento RS-485, em topologia *multi-drop*. Essa tecnologia torna o sistema flexível e aberto, facilitando a expansão de uma forma modular. A incorporação da tecnologia *Smart Card*, consente a implementação de serviços adicionais, tais como: registro de distância, consumo, velocidade máxima, nome do motorista, período de uso, etc.

KOJIMA et al. (1997) analisam as economias provocadas pela introdução de tecnologia de *Fieldbus* e examinam os instrumentos inteligentes de campo para as plantas de produção de óleo. Comparado com a instrumentação analógica convencional, o custo inicial dos dispositivos de campo, cablagem e engenharia foram reduzidos de 20 a 30 %, bem como a redução de custos de manutenção, devido à manutenção preventiva dos dispositivos de campo, e adquiridas significativas melhorias na disponibilidade do sistema.

RAMACHANDRAN et al. (1998) descreveram a tecnologia *Fieldbus*, usada para controlar um refinador de peróxido de hidrogênio, que permitia controlar processos distribuídos, com uma verdadeira interoperabilidade de vários fabricantes. A tecnologia aplicada colaborou para a otimização do brilho e a redução do uso de química. A instalação *Fieldbus Foundation* da *Daishowa* América foi também apresentada, incluindo a tecnologia *Fieldbus* instalada, resultados, e futuros planos.

SANTOS et al. (1999) tratam de estudos a respeito da automação de subestações de energia elétrica, em que foi realizado um levantamento dos principais dispositivos e sinais normalmente encontrados em uma subestação. A subestação da CESP, atualmente de propriedade da CPEE, em São José do Rio Pardo-SP, serviu como base para o desenvolvimento e análise de várias alternativas para se realizar um projeto de automação sendo utilizado o barramento de campo *Fieldbus*.

O sistema por si apresenta certas características intrínsecas, que influenciaram na escolha de um determinado tipo de topologia, meio físico, entre outros.

Considerações à respeito de reformas e construção de novas subestações foram

levadas em conta, quando a possibilidade de se implementar sistemas de automação diferentes foi colocada em questão.

Para finalizar, um sistema exclusivamente baseado em um ou mais barramentos de campo *Fieldbus* foi idealizado, o que se toma, a princípio, uma solução quase perfeita na área de subestações de energia elétrica.

VALERA et al. (1999) tratam do controle de um robô industrial, em uma rede de área local (LAN), usando um *Fieldbus* industrial: o PROFIBUS DP. Eles têm como principal objetivo explicar o uso dessa arquitetura para tornar disponível um controle distribuído e aberto. Também diferentes esquemas de controle avançado e suas respectivas implementações com um robô industrial são apresentadas, em que diferentes tipos de controladores e/ou referência geradores podem ser obtidos de uma maneira mais fácil.

Um conjunto inicial da *Fieldbus Foundation* de dez blocos funcionais padrão (FIELDDBUS FOUNDATION - 891, 1999), foram projetados para realizar tarefas básicas de aquisição, atuação, controle e monitoramento em malhas de controle de processos industriais. Posteriormente, mais onze blocos funcionais foram padronizados pela *Fieldbus Foundation* (FIELDDBUS FOUNDATION - 892, 1999), sendo esses, utilizados em aplicações mais avançadas e específicas de cálculo e controle (FIELDDBUS FOUNDATION - 892, 1999).

BLEVINS & PELUSO (1999), citam alguns exemplos de como controlar e monitorar aplicações que podem ser implementadas, usando esse novo conjunto de onze blocos funcionais padronizados. Algumas das aplicações consideradas são: caracterização do sinal, compensação dinâmica, cálculos, controle *split range*, regulador de válvula (*on - off*) e controle de motor.

FERRAZ (1999), propôs a implementação de um algoritmo de controle adaptativo, utilizando recursos *Fieldbus* para melhorar o controle e os resultados de processos não-lineares. Nesse sistema, o *Fieldbus* é utilizado somente como uma fonte de informação de dados ao algoritmo de controle adaptativo, que é implementado e executado no supervisor do computador central, realizando assim

o processamento centralizado da estratégia de controle. Ao contrario deste trabalho que propõe um processamento descentralizado da estratégia de controle nos dispositivos de campo.

Devido ao fato do processamento ser centralizado, várias desvantagens ocorrem, e algumas delas são citadas a seguir:

- a) O algoritmo de controle depende muito do tempo de processamento do computador;
- b) Perda do supervisor implica a perda do algoritmo de controle;
- c) Grande tráfego de dados na rede.

A figura a seguir ilustra o sistema de controle.

FIGURA 2.7 - *Fieldbus* usada apenas como uma fonte de informações dadas ao algoritmo adaptativo

HARROLD (1999), relata o controle baseado no modelo de referência para otimizar a eficiência de uma caldeira. O objetivo do projeto modelado para a otimização, foi explorar a viabilidade ao utilizar controle avançado para melhorar a eficiência da unidade da caldeira, reduzindo a emissão de NOx e outros gases de estufa, sem afetar a variação de carga da caldeira.

O *software Foxboro Connoisseur* foi utilizado para coleccionar e analisar dados, bem como construir o controlador multivariável, baseado no modelo de referência.

Implementado em um computador separado, o controlador multivariável foi conectado ao *loop* de controle do sistema de processo da Foxboro IA existente. Ver Fig.2.8.

FIGURA 2.8 - Diagrama de otimização da unidade da caldeira

O controlador multivariável ajustado e testado, foi conduzido em operação de malha aberta por mais de quatro dias. O ajuste final foi alcançado durante dois testes de verificação de dez horas, usando um perfil de carga pré programado com o sistema modelado para otimização, que operou em modo *off-line* e *on-line*.

Dados de teste de verificação revelaram melhorias na eficiência da unidade (caldeira, gerador, etc.) de 0.65%, reduções de NOx de 12.3% e a estimativa da redução do CO2 passou a ser de 51,500 toneladas por ano. Tudo alcançado sem qualquer degradação na variação de carga da caldeira.

2.5 TÉCNICAS AVANÇADAS (CONTROLE ADAPTATIVO)

O projeto de sistemas de controle de alto desempenho exige o uso de técnicas de controle adaptativo, quando os parâmetros do processo a serem controlados não são bem conhecidos ou variam durante uma operação normal.

Entre vários métodos alternativos, o uso da técnica conhecida como controle adaptativo por modelo de referência (CAMR), é um dos métodos mais viáveis, para uma possível implementação de sistemas de controle adaptativo. As técnicas de CAMR são geralmente classificadas em três métodos:

- (i) Controle adaptativo por modelo de referência indireto (CAMRI);
- (ii) Controle adaptativo por modelo de referência direto (CAMRD);
- (iii) Observador de estado adaptativo

2.5.1 MÉTODOS DE CONTROLE ADAPTATIVO

DUARTE & NARENDRA (1989), apresentaram uma nova abordagem de controle adaptativo usando a combinação dos métodos CAMRD e CAMRI em uma planta linear, desconhecida e invariante no tempo.

O erro de saída e os parâmetros observados da planta foram, simultaneamente usados na determinação dos parâmetros de controle para uma adaptação estável.

A estabilidade uniforme e global de todo o sistema foi construída através da seguinte função de *Lypunov*:

$$V = \frac{e_c^2 + |b_p| \phi_0^2 + |b_p| \phi_k^2 + c_i^2 + \eta_a^2 + \eta_b^2}{2} \quad (2.1)$$

Resultados foram apresentados para uma planta de primeira ordem com o objetivo de estabelecer os conceitos básicos envolvidos.

DUARTE & NARENDRA (1996), discutiram em detalhe um novo método de CAMRI para plantas lineares, invariantes e contínuas no tempo com parâmetros desconhecidos.

Nesse método, o erro de identificação dos parâmetros entre a saída da planta e a saída do modelo de identificação foi mostrado para ser limitado e tender para zero assintoticamente, no qual contrasta com o método de regulação adaptativo baseado na idéia do chamado “erro de equilíbrio de identificação”, proposto por KREISSELMEIER & SMITH (1986).

Nesse mesmo sentido, a prova de estabilidade do estado de equilíbrio de todo o sistema do novo CAMRI foi definido para ser diferente do que é usado no CAMRD. No entanto, a quantidade de informação exigida em ambos os casos, foi a mesma para obter um controle adaptativo estável.

Simulações de uma planta de segunda ordem, foram realizadas para comparar o desempenho do novo CAMRI com relação ao CAMRD. Um melhor comportamento transitório do esquema CAMRI foi observado.

MAKAÛUDI & RADOUANE (2000) apresentaram um método de CAMRI para sistemas de fase não mínima, de uma única entrada e uma única saída, com atraso desconhecido ou variante no tempo.

O esquema de observação de parâmetros foi obtido, através da combinação de uma filtragem adaptativa dos dados de entrada e saída, com um algoritmo de mínimos quadrados recursivo, que possui projeção de parâmetros e normalização de sinal.

Inicialmente foi aplicada a filtragem adaptativa dos dados e a seguir, o modelo filtrado foi então identificado pelo algoritmo de mínimos quadrados. Esse modelo filtrado e estimado de fase mínima, teve então os zeros realocados no interior de um círculo unitário e foi usado para a síntese do controle. Esse, consentiu tratar com sistemas de fase não-mínima e com dinâmicas não modeladas.

A filtragem adaptativa permitiu também, resolver a dificuldade do problema do CAMR para casos em que o atraso é desconhecido ou variante no tempo.

A vigorosidade do esquema à respeito da dinâmica não modelada e dos ruídos aditivos foi melhorada e pode ser vista com os resultados dos exemplos numéricos apresentados.

NONAKA et al. (1996) propuseram um novo controle de estrutura variável por modelo de referência, com superfície adaptativa deslizante, estimada diretamente da entrada e da saída do sistema, com vigorosidade para incertezas paramétricas.

Esse controle usou somente medições de entrada e saída e foi aplicado a um sistema linear invariante no tempo, de uma única entrada e uma única saída, com parâmetros desconhecidos.

Sob algumas suposições usuais no CAMR, o resultado do sistema de malha fechada foi mostrado para ser globalmente estável na presença de perturbações de entrada. Além disso, a atenuação da perturbação foi possível para algumas extensões, através de mudanças ocorridas nos parâmetros da superfície deslizante.

Foi também demonstrado através de estudos de simulações, que os efeitos das perturbações externas podem ser reduzidos pela estrutura do controlador, de modo deslizante, uma vez que, a identificação da função de transferência do erro do modelo adaptativo pode ser modificada, sem uma pré-filtragem, em que o grau

relativo, sendo de um menor grau do que o CAMR, permite que a forma de velocidade de convergência possa ser melhorada. Resultados das simulações foram apresentados para essas vantagens.

NARENDRA & ANNASWAMY (1989), apresentam uma lei adaptativa baseada na teoria de estabilidade de *Lyapunov*. A função quadrática de *Lyapunov* proposta é dada por:

$$V(e, \Phi, \Psi) = \frac{1}{2}e^2 + \frac{1}{2\gamma_1}|b_p|\left[\Phi^2 + \Psi^2\right] \quad (2.2)$$

E a lei adaptativa obtida é dada pelas equações:

$$\dot{\theta}_1 = \dot{\Phi} = -\gamma_1 e(t) x_p(t) \quad (2.3)$$

$$\dot{\theta}_2 = \dot{\Psi} = -\gamma_1 e(t) r(t) \quad (2.4)$$

Onde γ_1 é uma constante positiva, chamada ganho adaptativo, escolhido pelo projetista.

Também baseada na teoria de estabilidade de *Lyapunov*, RAO (1998) apresentou algumas modificações e melhoramento, com relação ao método de NARENDRA & ANNASWAMY (1989) e propôs uma nova função não-quadrática de *Lyapunov* dada por:

$$V(e, \Phi, \Psi) = \frac{1}{4}[e^4] + \frac{1}{2\gamma_3}|b_p|\left[\Phi^2 + \Psi^2\right] \quad (2.5)$$

Resultados das simulações produziram uma adaptação estável, com melhora na convergência comparada com a lei adaptativa padrão de NARENDRA & ANNASWAMY (1989).

McLAIN & HENSON (2000), propuseram uma estratégia de CAMR não linear, na qual um modelo linear (ou um conjunto de modelos lineares) é embutido dentro do controlador não linear para melhorar o desempenho de malha fechada durante a inicialização e nos transitórios.

A técnica é aplicada a um processo não linear de uma única entrada e uma única saída, com realimentação de estado completa e dinâmica de zero estável. A lei do controlador também não linear é construída, através do embutimento dos ganhos do controlador linear, obtidos do modelo linear ou dos múltiplos modelos lineares.

As funções do controlador de alta ordem são aproximadas com funções bases radiais, que são linearmente parametrizadas. O número de funções bases é determinado a priori e um algoritmo rodando *on-line*, é utilizado para assegurar que a função centrada próxima ao ponto operante corrente seja ativa.

Os parâmetros do controlador, que adapta a lei de controle e garante que a saída da planta siga, assintoticamente, a saída linear do modelo referência são deduzidos através da análise de estabilidade de *Lyapunov*.

Regras de combinação e estimação são usadas para embutir múltiplos modelos lineares, dentro do controlador não linear. Isso produz um novo esquema de controle adaptativo de múltiplos modelos não lineares, com potencial para produzir transições mais rápidas entre pontos operantes do que é possível com a técnica de controle adaptativo de múltiplo modelo linear.

A estratégia de controle proposta foi comparada com um controlador de realimentação de estado linear e um controlador adaptativo multi modelo, usando um modelo de reator químico não linear. Os resultados forneceram um desempenho regulatório e excelente rejeição à perturbações.

LI et al. (1996), estabeleceram técnicas de análises aplicáveis dos coeficientes local da Função de *Walsh* (FW) e usaram-na para adaptar os parâmetros *on line* do controlador no CAMR de um sistema de primeira ordem. A lei adaptativa é dada pelo método do gradiente, conforme demonstram as equações:

$$\dot{\theta}_1(t) = -\text{sgn}(b_p) \gamma e(t) x_p(t) \stackrel{\text{def}}{=} \beta_1(t) \quad (2.6)$$

$$\dot{\theta}_2(t) = -\text{sgn}(b_p) \gamma e(t) r(t) \stackrel{\text{def}}{=} \beta_2(t) \quad (2.7)$$

Para adaptar os parâmetros $\theta_1(t)$ e $\theta_2(t)$, o método dos coeficientes locais de *Walsh* foram usados, assumindo que os sinais do processo estão disponíveis e conectados a um gerador de vetor da FW, no qual produz os parâmetros dos coeficientes locais $\theta_{1,2}(t)$. Os vetores de coeficientes locais são ajustados pelo método do gradiente e as funções *Walsh*, como sinais paralelos, sendo então conectados a uma camada de coeficientes locais ajustáveis, multiplicado por esses coeficientes e somado para produzir a estimação de parâmetros desejada. De acordo com o algoritmo integral dos coeficientes locais de *Walsh*, os parâmetros ajustáveis podem ser representados da seguinte forma:

$$\delta\hat{\theta}_1(t_{j+1}) = (\delta\hat{\theta}_1(t_j) + \frac{1}{2^k} \delta\hat{\beta}_1(t_j) N) \quad (2.8)$$

$$\delta\hat{\theta}_2(t_{j+1}) = (\delta\hat{\theta}_2(t_j) + \frac{1}{2^k} \delta\hat{\beta}_2(t_j) N) \quad (2.9)$$

PHAN & FRUEH (1999), apresentaram um algoritmo de controle iterativo no qual o modelo de referência é usado para orientar o aprendizado do processo.

O modelo de referência fornece uma maneira, em que o conhecimento sobre o sistema (se disponível) é incorporado para produzir o aprendizado do processo com propriedades transitórias desejáveis.

Funções bases são usadas para construir um histórico dos dados de entrada e saída, para reduzir a dimensão do modelo do sistema e para promover o condicionamento no aprendizado do algoritmo.

Resultados experimentais de um sistema com dinâmica flexível foram usados para ilustrar o método.

MIYASATO (1998), propôs um método simplificado para sistemas de CAMR, utilizando observadores de alto ganho e leis adaptativas do tipo projeção.

Nesse método, a estrutura global do sistema de controle adaptativo não se torna tão complexa com o aumento do grau relativo do processo a ser controlado.

As condições de estabilidade do erro do sistema são, explicitamente reduzidas para a própria escolha de dois parâmetros de projetos, em que um parâmetro deverá ser suficientemente grande e o outro deverá ser suficientemente pequeno.

No caso ideal, quando nenhuma incerteza não estruturada existe, é mostrado que todo o sistema é globalmente limitado e a magnitude do erro de saída pode ser feita arbitrariamente pequena, através de ajustes em um dos dois parâmetros.

TSUJI et al. (1996) apresentaram um observador com controle adaptativo por modelo de referência. O projeto do novo controlador adaptativo não linear é aplicado para uma classe de sistemas de uma única entrada e uma única saída, em que a entrada é pré-multiplicada pelo produto de um vetor de parâmetros desconhecido e por um vetor de funções conhecidas não lineares. A nova lei adaptativa é definida pela seguinte equação:

$$\dot{\hat{\theta}}(t) = \dot{\hat{\theta}}(t) = \begin{cases} -\frac{\gamma \varepsilon(t) \Psi(t)}{1 + \Psi^T(t) \Psi(t)} & \text{para } |\varepsilon(t)| \geq \delta(t) \\ 0 & \text{para } |\varepsilon(t)| < \delta(t) \end{cases} \quad (2.10)$$

2.5.2 APLICAÇÕES DAS TÉCNICAS DE CONTROLE ADAPTATIVO COM MODELO REFERÊNCIA

NAJIM et al. (1995) dizem que alguns algoritmos de controle adaptativo são bem conhecidos e podem ser implementados de modo muito econômico. Na maioria dos casos, o controle adaptativo tem sido escolhido como estratégia de controle para processos não lineares, em que o controle linear tem se mostrado insuficiente para o uso. Muitas aplicações industriais, incluindo a engenharia química e de processo tem desempenhado muito nesse campo.

CHAO & NEOU (1999) utilizaram o controle adaptativo por modelo de referência (CAMR) de um *drive capstan* de ar-lubrificado, de forma que um único modelo fosse utilizado para minimizar a discrepância entre os modos macro e micro para resolver o problema de chaveamento dos modelos.

O método empregado para minimizar o erro entre a saída da planta e a saída do modelo foi o método do gradiente. A equação final dos parâmetros do controlador, após uma regra de ajuste para diminuir a dependência do ganho de

adaptação sobre os níveis de sinal, é dada pela seguinte expressão:

$$\theta' = \frac{\gamma \varphi e}{\alpha + \varphi^T \varphi} \quad (2.11)$$

Onde $\varphi = -\left(\frac{\partial e}{\partial \theta}\right)$ e $\alpha > 0$ é introduzido em casos onde φ é muito pequeno.

Os resultados do experimento de posicionamento, demonstraram que precisão melhor do que ± 15 nm com nenhum *overshooting* foi obtida em todas as condições testadas. Testes de perturbações da resistência do sistema também provaram ser satisfatórios.

MAHER et al. (1995), investigaram uma implementação experimental da técnica de controle adaptativa, baseada na estimação e no CAMR de um processo de fermentação de fluxo contínuo, no qual o objetivo do controle adaptativo foi o de regular e seguir a trajetória desejada da planta de concentração de substrato, determinada através da atuação na taxa de diluição.

A idéia básica utilizada foi combinar um algoritmo de estimação de parâmetros e um modelo baseado no estado específico, com o método de CAMR. Esse tipo de abordagem tem sido desenvolvido por DUARTE & NARENDRA (1989), resultando na combinação do esquema de controle adaptativo direto e indireto.

A lei de estimação adaptativa e a teoria de controle foram explicadas através de uma aplicação em uma planta piloto de fermento. O desempenho do controlador proposto baseado no método direto de *Lyapunov*, foi avaliado por uma aplicação real no processo de fermentação.

Resultados experimentais foram apresentados, mostrando o desempenho do sistema controle. As vantagens e as desvantagens do algoritmo em termos de seguir a regulação e o *overshoot* foram também mostradas.

SINHA & PECHEV (1999), desenvolveram o projeto de um controlador adaptativo por modelo de referência para veículos suspensos magneticamente usando o critério de máxima descida estável.

O algoritmo de adaptação é construído para reduzir o erro do espaçamento de ar entre o modelo referência e o sistema atual.

A relação explícita entre os parâmetros do desempenho do critério e a regra de adaptação de realimentação de estado foram deduzidas para um sistema de suspensão de um único grau de liberdade, baseados nos conceitos teóricos desenvolvidos por SINHA & HULME (1983).

Resultados experimentais de um pequeno aparelho de teste são apresentados para ilustrar a efetividade do controlador não linear, proposto na presença de variações de carga (massa suspensa), perturbações de força e *setpoint* do espaçamento de ar.

SILVA et al. (1997), aplicaram a estratégia de CAMR e estrutura variável ao controle de velocidade de um motor CC, excitado independentemente.

Simulações e resultados práticos foram apresentados, comprovando a eficiência dessa estratégia frente à perturbações na carga presente no eixo do motor e variações paramétricas.

2.6 JUSTIFICATIVA

Através das referências citadas anteriormente, deve ser destacado que nenhuma delas utiliza o controle avançado para realizar o controle distribuído em *Fieldbus*, devido ao fato de não existir nenhum bloco funcional para tratar desse problema.

Um problema mais complexo não tem sido tecnicamente resolvido a contento, por não poder se utilizar soluções modernas, através de espaço de estados em *Fieldbus* aplicados a controle adaptativo, estocástico e outros. No entanto, esse trabalho procura contribuir para as soluções desses problemas mais difíceis, apresentando uma solução adaptativa por espaço de estados, aplicadas as redes *Fieldbus*.

Essa proposta é original por ser uma linha de pesquisa que depende da ação de novos blocos funcionais específicos.

Capítulo 3

Redes *Fieldbus*

Neste capítulo são descritos os principais componentes da arquitetura do sistema *Fieldbus Foundation*, bem com as camadas de comunicação.

3.1 INTRODUÇÃO

O *Fieldbus* é uma rede de transmissão de dados para comunicação com equipamentos de instrumentação e controle de plantas industriais, tais como transmissores, atuadores e controladores, podendo, inclusive, ser utilizado em aplicações que requeiram especificações quanto aos requisitos de segurança intrínseca. Esta rede é do tipo digital, serial, *half-duplex* e *multi-drop*. Ela é digital porque as informações são transmitidas em forma de mensagens de acordo com as camadas de comunicação definidas pelo protocolo *Fieldbus*; serial, porque as informações são transmitidas e recebidas bit a bit; *half-duplex*, porque a comunicação é bidirecional, porém, em uma única direção a cada instante e *multidrop*, porque é permitida a comunicação entre vários equipamentos conectados à rede.

O *Fieldbus* surgiu com o objetivo de interligar e operar os instrumentos de campo com características diferentes e de diversos fabricantes. Usufruindo de toda sua inteligência através de uma rede, proporcionando a descentralização das tarefas. Esta interligação incorpora vantagens como: maior imunidade a ruídos, pré-

processamento em dados específicos, transmissão de informações adicionais dos dados capacitando o diagnóstico do dispositivo e a previsão de falhas, redução dos custos de projeto, de fiação, de instalação e de expansão, entre outras. A descentralização das tarefas é muitas vezes vista como uma possibilidade de espalhar entre vários dispositivos um determinado programa ou processo de controle na busca de melhor uso de suas características. Para alcançar uma maior confiabilidade foi prevista a capacidade de em caso de pane do dispositivo, sua substituição imediata por outro implementando o mesmo programa. Como os dispositivos podem ser diferentes e de diferentes fabricantes, a padronização das funções a serem distribuídas nos mesmos foi necessária. Estas funções são chamadas de Blocos Funcionais (BF). A interligação desses blocos funcionais é que define a estratégia de controle e programação do processo a ser controlado. Na configuração especifica-se a escolha do Bloco Funcional em que dispositivo será executado.

A versatilidade do *Fieldbus* permite, em caso de pane em alguns deles, a reconfiguração automática *online*, especificando em qual dispositivo(s) BF(s) do dispositivo em pane será(ão) executado(s). Sem esta padronização internacional a redundância de BF(s) (conseqüentemente a redundância de dispositivos) fica limitada a só ser implementada em dispositivos iguais e dos mesmos fabricantes. A verdadeira interoperabilidade e intercambialidade fica inviável.

O *Fieldbus* é um protocolo interoperável suportado pela quase totalidade dos fabricantes mundiais de instrumentação. Ao seu término deverá ter reconhecimento mundial, devido ao comprometimento destes fabricantes em seguir um padrão único. A opção de baixa velocidade para *Fieldbus* é 25 vezes mais rápida que os protocolos comuns para transmissores inteligentes, além de ser muito mais eficiente. Esta versão do *Fieldbus* foi projetada para usar o mesmo tipo de fiação dos transmissores analógicos e inteligentes, para facilitar a substituição do sistema. O *Fieldbus* é baseado no modelo OSI (*Open System Standards Organization*) para representar as várias funções requeridas em uma rede de comunicação.

O *Fieldbus* não é só mais um protocolo de comunicação digital. Ele foi concebido para a indústria de controle de processos de modo a atender plenamente a todos os itens de uma lista longa e antiga de desejos do usuário. Esta lista de desejos inclui itens tais como: - estar de acordo com o modelo ISO/OSI; - uso de cabos de

conexão de utilização industrial normal; - segurança intrínseca para atmosferas perigosas; - variáveis identificadas por *tags* e expressas em unidades de engenharia; - variáveis com status, onde o status indique as condições da variável; - Blocos de função, com parâmetros de entrada e saída padronizados, parâmetros de configuração padronizados e algoritmos padronizados.

3.2 ARQUITETURA DO SISTEMA *FIELD*BUS FOUNDATION

3.2.1 MODELOS DE COMUNICAÇÃO

A comunicação no *Fieldbus* pode ser entendida com referência no modelo de camadas OSI (*Open System Interconnection*). O *Fieldbus Foundation* é especificado de acordo com o modelo simplificado OSI, consistindo apenas em 3 camadas: Camada Física (*Physical Layer* (PHY)), Camada de Enlace de Dados (*Data Link Layer* (DLL)) e Camada de Aplicação (*Application Layer* (APL)). Na Fig. 3.1 pode ser visto um modelo OSI onde as camadas de 2 a 7 são implementadas na maior parte por software, e por essa razão muitas vezes são chamadas de “pilhas de comunicação.”

FIGURA 3.1 - Modelo OSI em comparação com modelo *Fieldbus*

O *Fieldbus Foundation* não especifica somente a comunicação, mas também alguns aplicativos de usuário, na qual o *Fieldbus Foundation* usa a comunicação, embora o modelo de OSI não especifique qualquer aplicativo de usuário.

A camada de aplicação do *Fieldbus Foundation* consiste em duas subcamadas: a Sub Camada de Acesso *Fieldbus* (*Fieldbus Access Sblayer* (FAS)) e a Especificação de Mensagem do *Fieldbus* (*Fieldbus Message Specification* (FMS)). A Sub Camada de Acesso especifica os serviços da Especificação de Mensagem do *Fieldbus* para a Camada de Enlace de Dados.

A estrutura das camadas, bem como os componentes que compõem a arquitetura *Fieldbus Foundation* são novamente mostrados pela Fig. 3.2.

FIGURA 3.2 - Arquitetura dos componentes do sistema *Fieldbus Foundation*

3.2.2 TOPOLOGIA DE REDE

A norma prevê uma topologia de rede bastante aberta, sendo que o usuário tem a liberdade para definir sua estrutura. Um ramo básico de rede (*link*) tem que ter no mínimo um dispositivo (*device*) do tipo *Link Master*, ou seja, um *device* cuja camada de enlace pode se tornar a LAS - *Link Active Scheduler*, ou o árbitro da rede, aquele

que envia mensagens de *token* e requisições de dados aos outros *devices*. O LAS organiza também a *live list* que é uma lista atualizada de todos os *devices* presentes naquele *link* da rede. Pode-se ter dois ou mais *link masters* em um *link*, sendo que neste caso apenas um terá o LAS, e os outros serão mestre *backup*, *devices* que poderão se tornar o LAS caso o mestre ativo sai da rede ou apresente alguma falha. Este é um dos mecanismos de redundância da rede. A Fig 3.3 mostra uma topologia *single link*.

FIGURA 3.3 - Topologia *single link*.

Outro tipo de topologia é a denominada *multi-link*. Neste caso vários *links* estão em comunicação, e a "ponte" entre elas é um *device* cuja camada de enlace é do tipo *bridge*. Este *device* tem vários MAUs ou *Medium Attachment Units* que são as portas que realizam as conexões ao meio físico, no caso o par trançado.

Um *device bridge* tem uma porta para o *link upstream* que é o *link* primário onde aquela *bridge* é configurada como *device* simples e no máximo 4 portas para 4 *links* secundários ou ramificações, para os quais a *bridge* tem que ser a *Link Master*.

Uma *bridge* tem algumas funções especiais para estabelecer a conexão entre os diferentes *links*. Entre as funções principais temos: encaminhamento de mensagens entre *devices* de *links* distintos, republicação de mensagens para os *links* secundários e sincronização da temporização da camada de enlace e da aplicação entre os *links*.

Neste caso a topologia é do tipo árvore. O usuário no caso deve estudar a melhor maneira de configurar a topologia: ligando os *devices* que, estão participando de uma mesma malha de controle em um mesmo *link* e assim aumentando o

desempenho do controle, mas perdendo em cablagem ou então ligando os *devices* de maneira a otimizar a cablagem, mas assim poderá ter de utilizar várias *bridges* e conseqüentemente perder em desempenho de controle, e em segurança. A Fig 3.4 mostra uma topologia *multilink*.

FIGURA 3.4 - Topologia *multilink*

3.2.3 PROCESSO DE APLICAÇÃO (AP)

Um processo de aplicação é definido como as informações mais os componentes de processamento de uma aplicação distribuída que pode ser acessada ou endereçada em uma rede. A norma *Fieldbus Foundation* prevê dois tipos de aplicação: as aplicações determinísticas ou *time critical* e as não determinísticas. Estas aplicações estão definidas nos *devices* como um conjunto de funções denominado *Application Process* (AP). Pode-se ter em um *device* mais de uma AP. A parte visível de uma AP para a rede é composta por objetos, que podem ser simples ou compostos, estes últimos são um conjunto de objetos simples. Como um exemplo de um objeto composto temos um *Function Block*. Os objetos de uma AP estão listados em um estrutura denominada *AP Directory*, que contém referências para todos os seus objetos visíveis pela rede com seus respectivos índices no Dicionário de Objetos (OD), além de um cabeçalho definindo sua estrutura. No caso de um objeto composto, a primeira linha deste objeto na *AP Directory* possui um cabeçalho definindo a sua estrutura. A Fig 3.5 ilustra estas relações.

FIGURA 3.5 - Estrutura dos Processo de Aplicação

3.2.3.1 FUNCTION BLOCKS APPLICATION PROCESS

Os *Function blocks Application Process* (FBAP) (Processo de Aplicação dos Blocos Funcionais) são APs constituídas de blocos funcionais. Os blocos funcionais são estruturas que realizam as funções básicas da malha de controle, como algoritmos de compensação, aquisição e atuação e são realizadas repetitivamente fechando assim as malhas de controle da aplicação.

Os blocos funcionais possuem parâmetros de configuração, de entrada e de saída além do algoritmo em si e são identificados e acessados nos dispositivos por um nome ou *tag* exclusivo, e estão integrados aos dispositivos.

Dependendo das características do bloco, este pode ser definido com *transducer*, *resource* ou bloco funcional.

Um bloco *resource* é utilizado para especificar as características gerais do dispositivo onde este se encontra, assim só é permitido um *resource* por dispositivo e como é um bloco de especificação, não possui parâmetros de entrada ou de saída.

O bloco *transducer* serve para desacoplar os outros blocos de operações específicas, de I/O nos diferentes sensores e atuadores, e é ele quem opera o *hardware* do dispositivo físico.

O bloco funcional é o que realiza as funções algébricas e algoritmos de controle, possui parâmetros de configuração e de entrada e saída. Sua execução é disparada repetidamente em intervalos de tempo fixo, e também são realizados ciclicamente pedidos para a publicação dos seus parâmetros de saída.

A *Foundation* normalizou um conjunto de blocos funcionais básicos e outros avançados, na qual são descritos no capítulo 6.

Uma descrição mais detalhada do Processo de Aplicação dos Blocos Funcionais também é apresentada no capítulo 6.

3.2.4 SISTEMA DE GERENCIAMENTO *KERNEL*

Os dispositivos físicos têm um AP especial que é responsável por fornecer um nível básico de interoperabilidade com outros dispositivos da rede. Este AP é conhecido como *System Management Kernel* – SMK (Sistema de Gerenciamento Kernel). Esse é especificado no *Fieldbus Foundation System Management*.

O SMK é representado no *Management VFD*. Este VFD é compartilhado pelo *Network Management Agent* (agente administrador da rede) do dispositivo. Um único VCR é usado para acessar *Management VFD*.

A arquitetura SMK especifica a estrutura interna do SMK e como ele interage com as entidades de gerenciamento do sistema externo. A interação dentro do dispositivo com o *Network Management Agent* e com o dispositivo APs é um assunto local e não é especificado.

Quando o dispositivo *Fieldbus* não contém um SMK, seu *tag*, seu endereço de rede, e certas informações SMIB devem ser construídas ou pré-configuradas através de alguns meios locais. Além de não poder receber o tempo de *clock* da aplicação sincronizada.

O SMK é uma entidade de gerenciamento de dispositivo responsável pela sincronização dos dispositivos da rede para seguir as instruções recebidas do sistema gerenciador. O *Systems Management Specification* define as interações do SMK com o sistema gerenciador, mas não especifica propriamente o sistema gerenciador.

Devido o sistema gerenciador não ser especificado, este pode estar completamente contido em um único dispositivo, ou pode estar distribuído entre os dispositivos. Quando distribuído, as partes do sistema e as interações não são especificadas.

O SMK realiza vários serviços para preparar e sustentar o dispositivo durante a operação da rede. A Fig. 3.6 ilustra esta operação. O SMK opera também como

Embora não mostrada na Fig. 3.6, o SMIB é definido no *Managemen* VFD OD e é suportado pelo diretório de SMK AP. O *Systems Management Specification* indica qual informação será escrita, e qual somente é de leitura. O SMPK também é usado para acessar algumas das informações como parte dos serviços SMK. O SMK pode tornar as informações do SMIB disponíveis para os APs locais através da interface local.

3.2.4.2.2 ATRIBUIÇÃO DE ENDEREÇOS E TAG

Antes que um dispositivo possa ser conectado ativamente a rede, a ele deve ser atribuído um nome e um endereço de link de dados. Os nomes dos dispositivos são identificadores específicos do sistema e são chamados de *tags* dos dispositivos físicos (*tags* PD). Uma exceção é o dispositivo temporário. Aos dispositivos temporários não são atribuídos *tags* e nem endereços. Eles simplesmente são conectados a rede à um dos quatro endereços do *link* de dados “visitante” reservado para eles na especificação do protocolo da camada de enlace de dados.

Os *tags* PD podem ser designados pelo fabricante, ou eles podem ser designados através do SMK, normalmente em um ambiente de trabalho *off-line*. O ambiente de trabalho *off-line* é preferido porque ele mantém os dispositivos sem *tags* fora da operação da rede.

O SMK para os dispositivos sem *tags* são configurados para o estado inutilizado e conectado ao barramento de um dos quatro endereços do dispositivo *default*. Os endereços do dispositivo *default* são especificados como endereços de nó não visitantes pela camada de enlace de dados.

O Escalonador ativo de conexões (*Link Active Scheduler* (LAS)) para um *link* que reconhece um novo dispositivo no endereço *default* adiciona ele a *Live List* de acordo com os procedimentos da camada de enlace. Como parte desta operação, o LAS mantém a *Live List*, a uma lista de todos dispositivos ativos no *link*.

A função do gerenciador do sistema responsável pelo *tag* e pela atribuição do endereço de *link* de dados é chamada de “Configuração *Master*”. Esta é normalmente colocada com o LAS, embora não se exija dela a monitoração da *Live List* para a adição de novos dispositivos. Quando ela encontrar um endereço *default*, ela verifica

que o SMK não tem um *tag* PD para o dispositivo, e atribui um usando o SMK_P e os procedimentos especificados no *Systems Management Specification*. Uma vez atribuído, o SMK move para o estado inicializado. Neste estado, o dispositivo está pronto para ser atribuído um endereço no nó de *link* de dados da operação da rede.

O procedimento de atribuição de endereço assegura que o SMK de um dispositivo na rede *Fieldbus* recebe um único endereço de nó na camada de enlace de dados. Antes de um endereço ser atribuído, somente o SMK é permitido para iniciar a transferência de dados no barramento e na camada de enlace de dados do dispositivo que é permitido para usar somente os endereços de dispositivo *defaults*.

A camada de enlace de dados de um dispositivo no estado inicializado unirá a rede exatamente da mesma maneira que ela atribuiu o *tag*. Depois do LAS adicionar isto para a *Live List*, a Configuração *Master* reconhece que ela uniu a rede e lê o *tag* de PD do SMK usando o SMK_I. Ela então checa para ver se o *tag* já está em uso no barramento.

Se o *tag* não está em uso, a Configuração *Master* atribui um endereço para o SMK do dispositivo. SMK escreve o endereço na camada de enlace de dados através de uma interface local. Uma vez isto ocorrido a Configuração *Master* instrui o SMK para transição do estado operacional. Como parte desta transição o SMK causa a camada de enlace de dados o movimento do dispositivo para este novo endereço de *link* de dados e espera que o reconhecimento do LAS para isto. Como parte deste procedimento, a camada de enlace de dados do dispositivo reconhecerá se o endereço já está em uso. Se ele estiver, o SMK notificará e reverterá de volta para o endereço *default*.

3.2.4.3 IDENTIFICAÇÃO DO DISPOSITIVO

O serviço de identificação SMK permite que os APs obtenham um *tag* PD e o dispositivo ID um SMK remoto. O Dispositivo ID é um sistema identificador independente que é fornecido pelo fabricante. Este serviço é também usado pela Configuração *Master* durante a atribuição do endereço para verificar que o dispositivo tem um *tag* e que o endereço correto é atribuído ao dispositivo.

3.2.4.4 LOCALIZAÇÃO REMOTA DOS DISPOSITIVOS E OBJETOS

O serviço de identificação SMK permite que os dispositivos APs transmitam uma pergunta para todos os SMKs na rede para acessar informações sobre um objeto remoto nomeado. Se um *tag* PD é questionado, o SMK contendo o *tag* PD responderá com o dispositivo ID. Caso contrário, todos os SMKs passam pelo questionamento de seus APs e esperam por uma resposta. O AP contendo o objeto nomeado responderá com o índice do objeto OD e os IDs do VCRs que podem ser usados para acessar o objeto. Uma vez que os nomes não estão duplicados na rede, somente um AP responderá.

3.2.4.5 SINCRONIZAÇÃO DO *CLOCK*

Cada *link* na rede *Fieldbus* contém uma *Application Clock Time Publisher* (Aplicação de Publicação do Tempo do *Clock*) responsável para distribuir a aplicação do Tempo no link. Uma e somente uma desta *Application Time Publisher* opera como uma fonte de aplicação. Isto origina as mensagens de aplicação do *clock* para a rede. Quando esta *Application Time Publisher* ou quaisquer outras falham em sua operação, procedimentos de recuperação do SMK são definidos automaticamente para selecionar e trocar um *Time Publisher* secundário localizado no mesmo *link*.

3.2.4.6 ESCALONAMENTO DO BLOCO FUNCIONAL

A sequência de operação dos blocos funcionais é determinada pelo *function block schedule* (escalonador de bloco funcional), na qual é uma tabela que registra quando cada bloco funcional deve ser disparado. Esta sequência é repetitiva e cada repetição realizada dentro de um período de tempo denominado macrociclo. Portanto os blocos funcionais são disparados pelo menos uma vez por macrociclo.

A Fig. 3.7 mostra o escalonamento de execução de blocos funcionais distribuídos entre dois transmissores.

FIGURA 3.7 - Escalonamento de blocos funcionais

Este macrociclo é quem determina a taxa de execução da malha de controle, pois esta é executada uma vez a cada macrociclo. Quanto maior o número de blocos na estratégia de controle, maior é o macrociclo.

3.2.5 SYSTEM MANAGEMENT KERNEL PROTOCOL COMMUNICATIONS (SMKP)

Para garantir o correto funcionamento do SMK, comunicações entre os SMKs são feitas usando o *System Management Information Protocol* (SMIK) (protocolo do núcleo do gerente do sistema). Esta comunicação ao invés de VCRs, utiliza diretamente os serviços da DLL, por meio de acesso direto a DLL. Estes pontos de comunicação DLL- SMK são denominados DLSAP ou *Data Link Service Access Point*.

Este tipo de comunicação pode ser ponto a ponto ou distribuída a um grupo de SMKs e é usada para mensagens de *assign addresses* (atribuição de endereços), *locate tags* (localização tags), *distribute time* (distribuição de tempo), *identify device* (identificação de dispositivos) e outras do contexto do SNK.

3.2.6 NETWORK MANAGEMENT AGENT

Este agente faz parte da estrutura de comunicação da norma denominada *Communications Entity*. Sua função é de garantir, para a rede o acesso as informações estatísticas e de configuração do denominado *communication stack* (pilha de comunicação) (conjunto formado pela FMS, FAS, DLL e PHY). O agente possui também uma base de dados composta por informações estáticas e dinâmicas que dizem respeito à basicamente três domínios:

- Informações referentes à entidade de comunicação como um todo,
- Informações referentes aos VCRs em uso e,
- Informações referentes aos protocolos de cada camada separadamente.

A *Network Management Agent* (NMA) (agente gerenciador de rede) acessa as informações das camadas diretamente através da *Layer Management Entity* (LMEs) (entidade gerenciadora de camadas). Como a comunicação é realizada de modo local, a norma deixa aberta a cada fabricante a arquitetura e as funções dos LMEs.

3.2.7 CAMADA DE APLICAÇÃO - FMS

Os serviços da Camada de Aplicação no *Fieldbus Foundation* são definidos pela *Fieldbus Messaging Specification* - FMS (Especificação de Mensagens *Fieldbus*). A FMS especifica os serviços e os formatos de mensagem para acessar os objetos do *Application Process* - AP (Processo de Aplicação) bem como as descrições do *Object Dictionary* - OD (Objeto Dicionário) para estes objetos. Os objetos visíveis da rede de um AP e suas correspondentes descrições do OD estão representados na FMS como um *Virtual Field Device* - VFD (Dispositivo de Campo Virtual). Os serviços específicos da FMS são definidos para acessar os atributos do VFD, assim como fabricantes e estados. A FMS, em combinação com descrições do OD, especifica a interface funcional para a aplicação dos dispositivos de campo.

Os serviços da FMS e o formato das descrições de objeto no OD são baseados nos tipos de objetos definidos na FMS, assim como variáveis e eventos. Para tornar os objetos de AP visíveis através da rede (acessíveis através dos serviços da FMS), estes devem ser representados usando um tipo de objeto da FMS.

Os serviços da FMS são fornecidos para os APs em VCR *Endpoint*. O VCR *Endpoint* representa o ponto de comunicação de um AP para um VCR. Os serviços disponíveis em um *endpoint* dependem do tipo do *endpoint*. Note que a FMS nunca executa o pedido de serviço, ela somente carrega os pedidos e as respostas entre as APs. Porém, nos Blocos de Função APs, muitos dos serviços são executados pela a de Bloco de Função Shell.

Quando um serviço da FMS é pedido, a FMS requer construir e enviar o pedido apropriado de mensagem a FMS em um remoto AP. A FMS propriamente não tem nenhum acesso para o OD. Por essa razão, é de responsabilidade do usuário da FMS codificar os dados de usuário para ser transferido na mensagem. Se o tipo de mensagem indica que uma resposta é para não ser retornada, o serviço não é confirmado. Se uma resposta é necessária, o serviço é confirmado. Os serviços confirmados sempre exigem o AP remoto para emitir uma resposta indicando se ela foi ou não capaz executar o serviço. Quando não for, o AP remoto responde retornando um código de erro. Estas mensagens são tecnicamente chamadas de FMS *Protocol Data Units* – PDUs (Unidades de Dados de Protocolo).

Os serviços confirmados são usados para manipular ou controlar os objetos AP. Por exemplo, eles são usados para ler e escrever o valor das variáveis. Eles são também usados para acessar o OD. Os serviços confirmados usam os VCRs *client/server* (cliente/servidor) para a troca de seus pedidos e respostas. Para suportar este tipo de VCR, a FMS fornece o controle de fluxo para os pedidos de serviço. Isto é, a FMS mantém um contador de pedidos submetidos para aquelas respostas que ainda não tem sido recebida, e para não aceitar os pedidos adicionais enquanto o contador manter um limite especificado.

Os serviços não confirmados são usados para publicar dados e para distribuir notificações de evento. Os dados publicados são carregados usando os VCRs *Publisher/Subscriber*. As notificações de evento são carregadas sobre os VCRs *Report Distribution* (Distribuição de Relatório). A diferença entre estes dois tipos de VCRs permite mostrar como a *Fieldbus Access Sublayer* - FAS (Subcamada de Acesso *Fieldbus*) usa a camada de link de dados para transferir as mensagens. A seguir é apresentada uma tabela com os resumos dos objetos FMS.

TABELA 3.1 - Serviços da FMS

Objeto da FMS	Serviço	Tipos de serviços	Significado	Tipo de VCR <i>Endpoint</i>
VCR <i>Endpoint</i>	<i>Initiate</i>	Confirmado	Iniciar uma conexão	Todos
	<i>Abort</i>	Não Confirmado	Terminar uma conexão	Todos
VFD	<i>Status</i>	Confirmado	E status de um <i>device</i>	Cliente /Servidor
	<i>Identify</i>	Confirmado	Ler nome do fabricante, tipo e versão	Cliente /Servidor
OD	<i>GetOD</i>	Confirmado	Ler a descrição de um objeto	Cliente /Servidor
	<i>PutOd</i>	Confirmado	Carregar a descrição de um objeto	Cliente /Servidor
<i>Domain</i>	Upload Segment	Confirmado	Pedir um bloco de dados	Cliente /Servidor
	Download segment	Confirmado	Enviar um bloco de dados	Cliente /Servidor
<i>Program Invocation</i>	<i>Start</i>	Confirmado	Iniciar um programa	Cliente /Servidor
	<i>Reset</i>	Confirmado	Ressetar um programa	Cliente /Servidor
<i>Variable List</i>	Write	Confirmado	Escrever uma variável	Cliente /Servidor
	<i>Physwrite</i>	Confirmado	Escrever em uma posição da memória	Cliente /Servidor
<i>Event</i>	<i>Event Notification</i>	Não Confirmado	Comunicar um evento	Report Distrib.
	<i>Alter Event Condition Monitoring</i>	Confirmado	Habilitar/Desabilitar um evento	Cliente /Servidor

3.2.8 SUBCAMADA DE ACESSO *FIELD*BUS – FAS

A *Fieldbus Access Sublayer* - FAS (Subcamada de Acesso *Fieldbus*) usa as características escalonadas e não escalonadas da camada de ligação de dados para fornecer um serviço a FMS. Os tipos de serviços da FAS são descritos pelo *Virtual Communication Relationships* - VCR.

O VCR tem as características semelhantes de um seletor da memória de um telefone. Existem muitos dígitos para discar em uma chamada internacional - um código de acesso internacional, um código do país, um código da cidade, um código de transferência, e um número de telefone específico.

As informações necessitam somente ser incorporadas apenas uma vez, e então o "numero a ser discado" é atribuído. Após a instalação, somente o número do seletor necessita ser incorporado para que a discagem ocorra. De uma forma similar, após a configuração, somente o número do VCR é necessário para se comunicar com um outro dispositivo do *Fieldbus*.

Os tipos de VCRs definidos para suportarem os mecanismos de transferência de dados, permitindo a interação de APs são:

3.2.8.1 VCR CLIENTE/SERVIDOR

VCR Cliente/Servidor rege as comunicações entre pares de processos de aplicação. Prove a interação de fluxo de dados bidirecional entre um único cliente e um único servidor baseado em mensagens enviadas diretamente entre APs. O cliente é aquele que inicia a comunicação. O servidor é aquele que executa a função correspondente ao pedido do cliente. O mecanismo cliente/servidor é ideal para o tráfego vertical na troca de informações entre os dispositivos de níveis diferentes. Mas ele não atende às restrições temporais relativas aos tráfegos horizontais do *Fieldbus*, nem considera as noções de gerenciamento da consistência de dados distribuídos nos diferentes dispositivos, justamente por fazer a comunicação entre somente dois dispositivos por vez. Mais de um processo em diferentes dispositivo pode necessitar de um mesmo dado produzido por um outro processo da aplicação de um outro dispositivo. Assim, neste caso ele teria que ser reenviado para estes dispositivos em instantes diferentes.

3.2.8.2 VCR PRODUTOR/CONSUMIDOR

O VCR Produtor/Consumidor (*Publisher/subscriber*) é o mecanismo de comunicação onde o fluxo de dados é unidirecional de uma entidade de aplicação para uma ou mais entidades de aplicação, baseados em dados/eventos envolvendo um único processo de aplicação produtor e um grupo de um ou mais processos de aplicação consumidores. Em outras palavras, onde toda variável pode ter seu valor alterado e ser transmitida para uma ou mais entidades da aplicação que recebem esta variável ao mesmo tempo. Deste modo, este mecanismo permite a mesma transmissão para vários processos, otimizando a ocupação do barramento por parte de uma mesma variável. O Produtor de um dado é a entidade da aplicação responsável pela produção deste dado e que o transmite. O consumidor de um dado é o processo de aplicação que, quando é executado, necessita dos dados recebidos. É, portanto, a entidade de aplicação o que recebe a variável.

Este mecanismo complementa o mecanismo cliente/servidor, a fim de atender às restrições dos tráfegos horizontais. Cada dispositivo associa a cada identificador que ele produz ou consome, um buffer para armazenar seu valor.

3.2.8.3 VCR DISTRIBUIÇÃO DE RELATÓRIOS

A transferência de dados sem conexão é caracterizada pela transferência enfileirada de DL-PDUs independentes dos endereços DL-SAP. Nenhum contexto é mantido na DLL para controlá-las. Este tipo de transferência é usado para Distribuição de Relatórios, pois o endereço de destino pode ser configurado como endereço DUSAP de Grupo, permitindo que múltiplos notificadoros de eventos enviem para o mesmo endereço de grupo, e permitindo que os receptores de evento sejam insensíveis à identidade e endereço destes notificadoros.

3.2.9 CAMADA DE ENLACE DE DADOS - DLL

A camada de enlace é um mecanismo para transferir dados de um nó da rede para os outros nós que necessitam de dados. Esta camada também administra a prioridade e ordem de tais pedidos de transferência de dados.

É de interesse da camada de enlace: dados, endereços, prioridades, controle ao meio, e outros que relacionaram a transferência de mensagem.

Considerando que a Camada de Enlace opera na baixa velocidade da Camada Física, essa tem mecanismos para usar um meio e um modo efetivo.

3.2.9.1 CONTROLE DE ACESSO AO MEIO

A funcionalidade mais importante da Camada de Enlace é o *Medium Access Control* (MAC) (Controle de Acesso ao Meio) do *Fieldbus*. Considerando que todos os dispositivos no mesmo barramento recebem o mesmo sinal da Camada Física, somente um deles é permitido transmitir o sinal de cada vez. O MAC é um método para alcançar este objetivo. O domínio dos dispositivos que compartilham o mesmo sinal da Camada Física é chamado de "*link*". Em outras palavras, somente um dispositivo do "*link*" é permitido usar o meio (Camada Física) de cada vez.

O *Link Active Scheduler* (LAS) (Escalonador ativo de conexões) tem o papel do MAC. Esse possui um *token* que o fornece a outro dispositivo para permitir enviar mensagens. O *token* é devolvido ao LAS para avançar o MAC.

Considerando que as mensagens de aplicação têm vários níveis de urgência, a Camada de Enlace suporta um mecanismo para transmitir as mensagens de acordo com a sua urgência.

A Camada de Enlace provê três níveis de "prioridade": *URGENT*, *NORMAL* e *TIME_AVAILABLE*, nesta ordem. Uma mensagem *URGENT* é transmitida imediatamente, mesmo que outras mensagens de *NORMAL* ou prioridade de *TIME_AVAILABLE* estejam na fila de espera, e vice-versa. O tamanho máximo dos dados permitido a cada prioridade é mostrado na Tabela 3.2.

TABELA 3.2 - Tamanho máximo dos dados em cada prioridade

Prioridade	Tamanho máximo dos dados
URGENT	64 bytes
NORMAL	128 bytes
TIME_AVAILABLE	256 bytes

3.2.9.2 ENDEREÇO

Na Camada de Enlace são identificados os parceiros de comunicação com o endereço-DL, na qual consiste em três componentes: *Link*, Nó e Seletor.

O campo *Link* consiste em 16 bits e identifica um "*link*". Quando a comunicação está dentro de um *link*, este campo é muitas vezes omitido. Este campo é necessário quando uma mensagem está sendo passada para outros *links* através das *bridges*.

O campo dá o endereço de nó de 8 bites. Um dispositivo *Fundation* tem um nó endereçado na extensão entre 0x10 e 0xFF que são classificados em *LM* (*Link Master*), *BASIC*, *Default* e *Temporary*. Normalmente os dispositivos estão em *LM* ou *BASIC* de acordo com sua classe de dispositivo. Quando um dispositivo perde o endereço de nó, ele se comunica usando um endereço *Default*.

Um dispositivo *Temporary* como um comunicador de segurança tem endereço de nó *Temporary*. O LAS tem um endereço de nó de 0x04.

Há um espaço de endereço de tamanho V(NUN). Se um dispositivo tem um

endereço neste espaço, este nunca unirá a um link.

V(FUN) e V(NUN) são parâmetros que podem ter acesso através do administrador da rede.

O campo *seletor* dá um endereço no dispositivo-interno de 8 bits para identificar uma *Virtual Communication Relationship* (VCR) (Relação de Comunicação Virtual).

Quando uma VCR é conectada a outra VCR, isto é identificado com o *Data Link Connection End Point* (DLCEP) (Ponto Final da Conexão de Link de Dados) mostrado neste campo.

Quando uma VCR não é conectado a outra, mas é aberto para enviar ou receber mensagens, isto é identificado com o *Data Link Connection End Point* (DLSAP) (Ponto de Acesso de Serviço de Link de Dados) mostrado neste campo. O DLCEP e o DLSAP têm diferentes extensões.

Alguns endereços DL são reservados para propósitos específicos. Por exemplo, os dispositivos podem compartilhar o mesmo "DLSAP global" para recepção de alarme.

3.2.9.3 LINK ACTIVE SCHEDULER - LAS

O *Link Active Scheduler* (LAS) (Escalonador ativo de conexões) tem o papel do MAC. Os dispositivos *Foundation* são classificados como: *BASIC*, *Link Master* (LM) e *Bridge*. Uma classe LM tem a capacidade para trabalhar como o LAS, enquanto os dispositivos de classe *BASIC* não têm essa capacidade. Um dispositivo de classe *Bridge* possui além da capacidade de LM, a funcionalidade para conectar *links*.

Somente um dispositivo dentro do *link* trabalha como o LAS. Então pelo menos uma classe de dispositivo LM é necessária dentro de um *link*.

Os dispositivos de LM tentam adquirir o papel do LAS, quando nenhum LAS existe na partida, ou quando o corrente LAS falha. O dispositivo LM com o menor endereço de nó ganha esta contenção. Outros dispositivos LM observam a atividade do LAS e assumem sobre ele seu papel, quando LAS vai para longe.

3.2.9.4 COMUNICAÇÃO ESCALONADA

A transferência de dados escalonada é tipicamente usada para a transferência cíclica do *loop* de dados do processo entre os dispositivos do *Fieldbus*. As transferências usam o tipo *publisher/subscriber* para relatar a transferência de dados. O LAS é responsável para escalonar a comunicação e mantém uma lista do tempo transmitido para todos os *publisher* em todos os dispositivos que precisam ser ciclicamente transmitidos. Quando for a hora de um dispositivo publicar os dados, o LAS emite a mensagem *Compel Data* (CD) para o dispositivo. Após a recepção do CD, o dispositivo “publica” os dados para todos os dispositivos no *Fieldbus*. Os dispositivos que são configurados para receber os dados são chamados de “*Subscribers*” e os dispositivos que são configurados para enviar os dados são chamados de “o “*Publishers*”.

3.2.9.5 COMUNICAÇÃO NÃO ESCALONADA

A transferência de dados não escalonada é usada para as mudanças inicializadas pelo usuário, mudanças de modos (auto, man , etc), mudanças de ajuste, mudanças de *set point* e *upload/download*. A transferência não escalonada usa a distribuição de relatório ou os relatórios do tipo Cliente/Servidor para transferir dados. Todos os dispositivos no *Fieldbus* recebem uma chance para enviar as mensagens não sincronizadas entre as transmissões de dados escalonados. O LAS concede a permissão para um dispositivo usar o *Fieldbus* para emitir uma mensagem *Pass Token* (PT) para o dispositivo. Quando o dispositivo receber o PT, é permitido enviar as mensagens até que esse termine ou até o “*maximum token hold time*” tenha expirado.

3.2.9.6 MANUTENÇÃO DO LINK

Um outro papel do LAS é para manter o link. O LAS fornece um *token* para todos os dispositivos detectados. Quando um novo dispositivo é adicionado a rede, deve ser reconhecido pelo LAS e entrar para *Live List*.

3.2.10 CAMADA FÍSICA

A camada física é um mecanismo utilizado para transmitir e receber sinais elétricos ou óticos de forma que os dados consistindo de “zeros” e “uns” são transmitidos de um nó para outros nós. Esta pode ser de diferentes tipos de acordo com as características e ambiente de aplicação da rede.

Na camada física é analisado os tipos de ligações possíveis (fiação, cabos coaxiais, ótico ou rádio), conexões, terminadores, características elétricas, etc... especificados pela *Fieldbus Foundation* (FIELD BUS FOUNDATION PHYSICAL LAYER PROFILE SPECIFICATION - 816, 1995).

Embora o padrão IEC/ISA especifique vários meios de comunicação com várias velocidades, o *Fieldbus Foundation* escolheu seu próprio subconjunto, ou seja, meios de comunicação de fibras e cabos de baixa velocidade e Ethernet.

3.2.11 CAMADA DE USUÁRIO

A Camada de Usuário da *Fieldbus Foundation* define a completa distribuição das funções de aquisição de dados e controle nos dispositivos de campo, através de blocos de controle, estrutura desses controles e desses blocos funcionais, assim como, define a estrutura da base de dados que residirá em cada dispositivo de medição e controle.

É através desta camada que o usuário poderá ver o mundo da rede *Fieldbus* e seus dispositivos de campo. E com a padronização adequada dessa camada, poder-se-á operar no mesmo sistema, com múltiplos dispositivos independentemente de seu fabricante, sem perda da funcionalidade. Isto é conhecido como interoperar e a total interoperabilidade será obtida pela combinação do Dicionário de Objetos (OD), das Descrições do Dispositivo/*Device Descriptions* (DD) e dos Blocos Funcionais.

3.2.11.1 DICIONÁRIO DE OBJETOS (OD)

O Dicionário de Objetos de cada dispositivo físico é composto das informações mínimas sobre os objetos de comunicação contidos nesse dispositivo, para que não

ocupem muito espaço nos dispositivos. Alguns de seus objetos tem parâmetros com ponteiros que apontam para outros campos nas Descrições do Dispositivo, especificando propriedades dos parâmetros específicos.

3.2.11.2 DESCRIÇÃO DO DISPOSITIVO - *DEVICE DESCRIPTION* (DD)

A Descrição do Dispositivo é um mecanismo para descrever qualquer função nova que se queira adicionar nos dispositivos de campo, sem necessidade de atualizações, nem de revisões de *software* do sistema *host*, nem de *drivers* específicos, usufruindo das vantagens da evolução e inovação destes dispositivos.

Para isto, as descrições do dispositivo são desenvolvidas pelo fabricante do dispositivo com DDL - *Device Description Language*, como uma extensão do Dicionário de Objetos, mantido pelo FMS - *Fieldbus Messaging Specification*. O DDL descreve um determinado dispositivo de campo de forma não ambígua. Basta que estes usuários carreguem as especificações da comunicação do novo dispositivo, armazenadas em DDL, no seu sistema *host*. Deste modo, é passado para um *host* todas as características deste dispositivo, seus parâmetros e seu modo de operação. Assim qualquer *host* pode se comunicar com ele. Sem estas descrições seria impossível reconhecer as informações e operações que estariam dentro de cada dispositivo. Mesmo que se especificassem vários itens, com a evolução tecnológica, haveria necessidade de se atualizar a base de dados de todos os sistemas para que aceitassem um dispositivo com novas características. Com o DDL, especificou-se como passar as informações sobre estas características. Sem o DDL, só se enxergariam as características pré-definidas e que todos os dispositivos teriam.

Capítulo 4

Teoria de Controle Moderna

Este capítulo aborda a representação por espaço de estado e algumas técnicas empregadas na observação de estados de sistemas multivariáveis.

4.1 INTRODUÇÃO

A tendência atual dos sistemas de engenharia é de se tornarem cada vez mais complexos, isto é, serem variantes no tempo, possuírem múltiplas entradas e múltiplas saídas, e estas estarem inter-relacionadas de maneira complexa, devido principalmente às necessidades de realizarem-se tarefas difíceis e com requisitos de boa precisão.

O funcionamento adequado na maioria dos processos industriais exige que mais de uma variável do processo seja controlada simultaneamente. Como consequência, podemos encontrar mais de uma malha de controle para controlar um determinado processo. Estes sistemas de controle são denominados multivariáveis, na qual possuem uma das duas formas mais comumente encontradas em controle de processo:

- a) Os que controlam variáveis distintas inter-relacionadas de um mesmo processo, através de malhas de controle fisicamente separadas para cada variável;
- b) Os que controlam variáveis distintas inter-relacionadas dos processos como malha de controle que não são independente fisicamente, mas interagem devido à presença de instrumentos ou dispositivos comuns nos circuitos.

Desta forma, este capítulo aborda o projeto de controle de sistemas multivariáveis empregando as técnicas de controle moderno.

4.2 TEORIA DE CONTROLE CLÁSSICA

O modelo matemático de um sistema é definido como um conjunto de equações para representar o sistema físico. A maioria dos sistemas físicos podem ser caracterizados por equações diferenciais. Uma equação diferencial de ordem n representando um sistema pode ser escrita da seguinte forma:

$$y^{(n)} + a_1 y^{(n-1)} + \dots + a_{n-1} \dot{y} + a_n y = b_0 u^{(n)} + b_1 u^{(n-1)} + \dots + b_{n-1} \dot{u} + b_n u \quad \text{eq.(4.1)}$$

Onde y é chamada de variável de saída e u é a variável de entrada. Aplicando a Transformada de Laplace na eq.(4.1) e considerando-se nulas todas as condições iniciais, podemos definir a Função Transferência (FT) deste sistema através da relação da saída pela entrada dada por:

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \dots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n} \quad \text{eq.(4.2)}$$

A FT é a denominação dada ao modelo matemático no domínio da variável complexa s que relaciona uma variável dependente de resposta ou saída com uma variável de estímulo/perturbação ou entrada.

A teoria de controle clássica é baseada nos conceitos da FT e geralmente é empregada na análise e projeto de sistemas de uma entrada e uma saída. Quando empregada em sistemas de controle com uma entrada e uma saída fornece resultados satisfatórios. No entanto, não podem manipular os sistemas de controle com entradas múltiplas e saídas múltiplas.

Uma alternativa para solucionar esse problema é utilizar a teoria de controle moderno, que se baseia nos conceitos de espaço de estados e utiliza uma análise vetorial-matricial. Essa abordagem é apresentada a seguir.

4.3 TEORIA DE CONTROLE MODERNO

O objetivo do modelo de variáveis de estado, ou modelo de espaço de estados, é desenvolver uma representação, que preserve a relação entrada-saída (a mesma da Função de Transferência), mas que seja expressa em n equações de primeira ordem para um sistema de ordem n . A vantagem de n equações de primeira ordem é que, além das características entrada-saída, as características internas do sistema são representadas. Outras razões para o desenvolvimento dos modelos de estado são as seguintes:

a) Análise e projeto dos modelos de estado, auxiliados por computador são executados mais facilmente para os sistemas de ordem elevada, enquanto a abordagem pela função de transferência tende a falhar para esses sistemas.

b) Nos procedimentos de projeto por variáveis de estado, é realimentado um maior número de informações (variáveis internas) sobre o processo, assim podemos implementar um controle com mais informações para o sistema, que o elaborado a partir do método da Função de Transferência.

c) Os procedimentos de projeto, que resultam no melhor sistema de controle são quase todos baseados nos modelos de variáveis de estado. Por melhor sistema devemos entender que ele foi projetado de forma a minimizar (ou maximizar) a função matemática que expressa o critério de projeto.

d) Mesmo se não implementarmos o projeto por variáveis de estado, podemos produzir uma boa resposta do sistema. Então procuramos nos aproximar dessa boa resposta, usando o método clássico de procedimento de projeto.

e) Os modelos de variáveis de estado geralmente são necessários para as simulações (soluções de equações diferenciais pelo computador).

4.3.1 MODELAMENTO POR VARIÁVEIS DE ESTADO

O sistema linear representado pela equação diferencial de ordem n , eq.(4.1), pode ser representado por espaço de estado através das seguintes equações matriciais:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix} u \quad \text{eq.(4.3)}$$

$$y = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \beta_0 u \quad \text{eq.(4.4)}$$

Onde:

$$\begin{aligned} \beta_0 &= b_0 \\ \beta_1 &= b_1 - a_1 \beta_0 \\ \beta_2 &= b_2 - a_1 \beta_1 - a_2 \beta_0 \\ \beta_3 &= b_3 - a_1 \beta_2 - a_2 \beta_1 - a_3 \beta_0 \\ &\vdots \\ \beta_n &= b_n - a_1 \beta_{n-1} - \cdots - a_{n-1} \beta_1 - a_n \beta_0 \end{aligned} \quad \text{eq.(4.5)}$$

As eqs.(4.3 e 4.4) podem ser reescritas através de uma forma padrão, na qual são dadas por:

$$\dot{x}(t) = A x(t) + B u(t) \quad \text{eq.(4.6)}$$

$$y(t) = C x(t) + D u(t) \quad \text{eq.(4.7)}$$

Onde o vetor, $\dot{x}(t)$, é a derivada no tempo do vetor $x(t)$. Nessas equações condições:

A, B, C, D = Matrizes de parâmetros;

A = matriz (n x n), chamada matriz do sistema;

B = matriz (n x r), chamada matriz de entrada;

C = matriz (p x n), chamada matriz de saída;

D = matriz (p x r) que representa o acoplamento direto entre entrada e saída;

$x(t)$ = vetor dos estados = vetor $(n \times 1)$ dos estados de um sistema de ordem n ;

$y(t)$ = vetor de saída = vetor $(p \times 1)$ composto pelas saídas definidas;

$u(t)$ = vetor de entrada = vetor $(r \times 1)$ composto pelas funções de entrada do sistema.

A eq.(4.6) é chamada de equação de estado, e a eq.(4.7) é a equação de saída. A equação de estado é uma matriz de equações diferenciais de primeira ordem, e o vetor de estado, $x(t)$, é sua solução. Tendo conhecimento de $x(t)$ e do vetor de entrada $u(t)$, a equação de saída produz a saída $y(t)$.

Uma representação por diagrama de blocos das eqs.(4.6 e 4.7) pode ser visto através da Fig. 4.1.

FIGURA 4.1 - Diagrama de blocos da equação de estado e de saída

Normalmente, a matriz D é zero, já que nos sistemas físicos aparece uma dinâmica em todos os caminhos entre as entradas e saídas. Logo, as equações por espaço de estado podem ser escritas da seguinte forma:

$$\dot{x} = A x + B u \quad \text{eq.(4.8)}$$

$$y = C x \quad \text{eq.(4.9)}$$

Uma representação por diagrama de blocos das eqs.(4.8 e 4.9) pode ser visto através da Fig. 4.2.

FIGURA 4.2 - Diagrama de blocos da equação de estado e de saída (matriz D nula)

A equação característica do sistema que determina as localizações dos pólos é obtida através da seguinte equação:

$$\det([sI - A]) = 0 \quad \text{eq.(4.10)}$$

Onde I é a matriz identidade de dimensão n .

4.3.2 LEI DE CONTROLE

O primeiro passo no projeto de espaço de estados é obter uma lei de controle como realimentação de uma combinação linear das variáveis de estado, na qual pode ser dada por:

$$u = -Kx = -[K_1 \quad K_2 \quad \dots \quad K_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{eq.(4.11)}$$

É admitido aqui que todos os elementos do vetor de estado estejam disponíveis para aquisição. Na prática, em muitos sistemas de controle é fisicamente ou economicamente inviável instalar transdutores necessários para medir todos os estados.

Através da eq.(4.11) temos a informação de que o sistema tem uma matriz ganho constante no ramo da realimentação do vetor de estados, assim como mostra a Fig. 4.3.

FIGURA 4.3 - Sistema com a matriz ganho constante na realimentação

Para um sistema de ordem n , existirão n ganhos de realimentação $(K_1, K_2, \dots, K_n)$, pois o sistema terá n raízes. Isso é possível para que exista um grau suficiente de liberdade para selecionarmos arbitrariamente qualquer localização desejada das raízes através da escolha dos próprios valores de K_i .

Substituindo a lei de realimentação dada pela eq.(4.11) no sistema descrito pela na eq.(4.8) temos:

$$\dot{x} = A x + B [-K x]$$

$$\dot{x} = (A - B K) x \quad \text{eq.(4.12)}$$

Aplicando o determinante na eq.(4.12), obtemos a equação característica do sistema de malha fechada, na qual calculamos os autovalores do sistema:

$$\det [sI - (A - B K)] = 0 \quad \text{eq.(4.13)}$$

Fazendo uma análise na eq.(4.13), podemos observar que essa equação produz um polinômio em s contendo os ganhos $K_1, K_2, \dots, K_n$. Logo, o projeto da lei de controle consiste então em ajustar os ganhos K de forma que as raízes da equação estejam na localização desejada. Por exemplo, assumindo que as localizações desejada dos pólos são $s = s_1, s_2, \dots, s_n$, logo a equação característica se torna:

$$\alpha_c(s) = (s - s_1)(s - s_2) \cdots (s - s_n) = 0 \quad \text{eq.(4.14)}$$

Os elementos exigidos de K são obtidos igualando os coeficientes das eqs.(4.13 e 4.14). Isso força a equação característica do sistema para ser idêntica à equação característica desejada, e conseqüentemente os pólos de malha fechada podem ser colocados nas localizações desejadas.

Quando a ordem do sistema é maior do que três, as computações envolvidas podem se tornar muito tediosas. Uma alternativa para resolver problemas deste tipo poder ser obtida através da formula de *Ackermann*, FRANKLIN (1994).

4.3.2.1 INTRODUÇÃO DA ENTRADA DE REFERÊNCIA COM REALIMENTAÇÃO DE ESTADO DE ORDEM PLENA

Uma forma de analisar a resposta transitória em projetos de localização de pólos é introduzindo uma entrada de referência no sistema. Uma maneira de realizar isto é mudar a lei de controle da pela eq.(4.11) para:

$$u = -Kx + r \quad \text{eq.(4.15)}$$

No entanto, o sistema agora terá um erro em regime permanente para uma entrada degrau. Um modo para corrigir esse problema é computar os valores em regime permanente dos estado e da entrada de controle, que resulta um erro de saída zero e então força-los a tomarem esses valores. Se os valores finais desejados dos estados e da entrada de controle são x_{rp} , e u_{rp} , então a nova equação de controle deve ser:

$$u = u_{rp} - K(x - x_{rp}) \quad \text{eq.(4.16)}$$

Desta forma, quando $x = x_{rp}$ (erro zero), $u = u_{rp}$. Para ajustar o valor final correto, deve-se resolver as eqs.(4.6 e 4.7) de forma que o sistema tenha um erro em regime permanente igual a zero para qualquer entrada constante. Logo, nessas condições temos:

$$0 = A x_{rp} + B u_{rp} \quad \text{eq.(4.17)}$$

$$y_{rp} = C x_{rp} + D u_{rp} \quad \text{eq.(4.18)}$$

Resolvendo as eqs.(4.17 e 4.18) para os valores na qual $y_{\text{tp}} = r_{\text{tp}}$ para qualquer valor de r_{tp} fazemos $x_{\text{tp}} = N_x r_{\text{tp}}$ e $u_{\text{tp}} = N_u r_{\text{tp}}$. Com estas substituições as eqs.(4.17 e 4.18) podem ser escritas por uma equação matricial:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{eq.(4.19)}$$

Resolvendo esta equação para N_x e N_u , temos:

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{eq.(4.20)}$$

Com estes valores finalmente a entrada de referência pode ser introduzida de forma a se obter um erro zero em regime permanente para uma entrada degrau:

$$u = -K x + (N_u + K N_x) r \quad \text{eq.(4.21)}$$

O coeficiente $N = N_u + K N_x$ é uma constante que pode ser antecipadamente computada. Logo:

$$u = -K x + N r \quad \text{eq.(4.22)}$$

Um diagrama de bloco do sistema é mostrado através da Fig.4.4.

FIGURA 4.4 - Diagrama de blocos de um sistema com realimentação de estados e entrada de referência

4.3.3 OBSERVADOR DE ESTADOS DE ORDEM PLENA

A lei de controle descrita no item 4.3.2, supõe todos os estados disponíveis para realimentação, ou seja, que todas as componentes do vetor de estado podem ser medidas. Infelizmente esse não é o caso na maioria das situações, devido ao custo dos sensores exigidos ou por ser impossível, fisicamente, medir todas as variáveis de estado. Então é necessário observar as variáveis de estado não disponíveis através de uma estimativa $\hat{x}$, do estado verdadeiro x , por algum mecanismo conhecido como observador, e substituir o estado verdadeiro pela sua estimativa na lei de controle.

Um observador de estados estima as variáveis de estado, baseado nas medidas das variáveis de saída e de controle. Se todas as variáveis de estado do sistema são observadas pelo observador de estado, independentemente se algumas variáveis de estado estão disponíveis para medição direta, é chamado de observador de estado de ordem plena. Seja a seguinte planta definida pelas equações abaixo:

$$\dot{x} = A x + B u \quad \text{eq.(4.23)}$$

$$y = C x \quad \text{eq.(4.24)}$$

Um método de observar os estados é construir um modelo de ordem plena da dinâmica da planta, isto é:

$$\dot{\hat{x}} = A \hat{x} + B u \quad \text{eq.(4.25)}$$

$$\hat{y} = C \hat{x} \quad \text{eq.(4.26)}$$

Onde $\hat{x}$ é estimativa do estado atual x . São conhecidos A , B , C e $u(t)$. Este observador de estados será satisfatório se for possível obtermos uma condição inicial correta de $x(0)$ e um conjunto de $\hat{x}(0)$ igual a isto. A Fig. 4.5 mostra um observador de estados de malha aberta que representa esta dinâmica.

FIGURA 4.5 - Observador de estados de malha aberta

Porém, é justamente a falta de informações sobre $x(0)$, que exige a construção de um observador de estados. Caso contrário, o observador de estado seguiria exatamente o estado verdadeiro. Deste modo, se realizada uma estimativa pobre para a condição inicial, o observador de estado terá um erro continuamente crescente ou um erro que vai muito lentamente para zero. Além disso, erros pequenos no sistema (A, B, C) causaria também uma estimativa para divergir do estado verdadeiro.

Para estudar a dinâmica desse observador, vamos definir o erro na estimativa para ser a diferença entre o estado verdadeiro x e o estado observado $\hat{x}$, assim como mostra a eq.(4.27):

$$e \triangleq x - \hat{x} \quad \text{eq.(4.27)}$$

Então a dinâmica desse sistema de erro é dada por:

$$\dot{e} = A e \quad \text{eq.(4.28)}$$

$$e(0) = x(0) - \hat{x}(0) \quad \text{eq.(4.29)}$$

O erro converge para zero para um sistema estável (A estável), mas não possui nenhuma capacidade de influenciar a taxa na qual o estado observado converge para o estado verdadeiro. Além disso, o erro está convergindo para zero na mesma taxa da dinâmica natural de A. Se essa taxa de convergência fosse satisfatória, nenhuma técnica de controle ou observação seriam exigidas.

Agora vamos realimentar o sistema com a diferença entre as saídas medidas e as observadas e corrigindo o modelo continuamente com esse sinal de erro. A equação para esse esquema, mostrado na Fig. 4.6, é:

$$\dot{\hat{x}} = A \hat{x} + B u + L (y - C \hat{x}) \quad \text{eq.(4.30)}$$

FIGURA 4.6 - Observador de estados de malha fechada

Onde L é um ganho proporcional definido como:

$$L = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} \quad \text{eq.(4.31)}$$

O ganho L é escolhido para realizar satisfatórias características do erro. A dinâmica do erro pode ser obtida subtraindo a estimativa da eq.(4.30) do estado da eq.(4.23), para se conseguir a equação de erro:

$$\dot{e} = (A - LC)e \quad \text{eq.(4.32)}$$

A equação característica do erro é obtida a seguir por:

$$\det[sI - (A - LC)] = 0 \quad \text{eq.(4.33)}$$

Se escolha de L , for de forma que $(A - LC)$ tenha autovalores razoavelmente rápidos e estáveis, então o erro decairá para zero e permanecerá lá independente da conhecida entrada $u(t)$ e seu efeito sobre o estado $x(t)$ independente da condição inicial $x(0)$. Isso significa que $\hat{x}(t)$ convergirá para $x(t)$, não importando o valor de $\hat{x}(0)$. Além disso, podemos escolher a dinâmica do erro para ser estável, bem como mais rápida do que a dinâmica de malha aberta determinada por A .

Deve ser notado, que em se obter a eq.(4.32) é assumido que A , B , e C são idênticos à planta física e a implementação do observador. Se o modelo da planta (A , B , C) não for preciso, a dinâmica do erro não é mais governada pela eq.(4.32). Porém, é permitido escolher L de forma que o erro do sistema seja pelo menos estável e o resíduo do erro seja aceitavelmente pequeno, mesmo com erros de modelagem e entradas perturbadoras.

Dessa forma, a seleção de L pode ser aproximada da mesma maneira como K é selecionado no projeto da lei de controle. Isto é, especificando a localização desejada dos pólos do erro do observador como $s = \beta_1, \beta_2, \dots, \beta_n$, e então, equação característica desejada do observador passa a ser:

$$\alpha_c(s) \triangleq (s - \beta_1)(s - \beta_2) \cdots (s - \beta_n) \quad \text{eq.(4.34)}$$

Logo, L pode ser calculado comparando os coeficientes das eqs. (4.33 e 4.34).

4.3.4 PROJETO DO COMPENSADOR: COMBINAÇÃO DA LEI DE CONTROLE E OBSERVADOR

Combinando a lei de controle do item 4.3.2 com o observador do item 4.3.3 e implementando a lei de controle usando as variáveis de estado observadas, obtemos o mecanismo de um controlador e observador, assim como mostrado na Fig 4.7.

Desta forma, esta dinâmica pode ser descrita através da equação da planta com realimentação da seguinte forma:

$$\dot{\hat{x}} = A \hat{x} - BK \hat{x} \quad \text{eq.(4.35)}$$

Esta equação pode ser reescrita em termos da equação de estado do erro ($e = x - \hat{x}$). Assim:

$$\dot{\hat{x}} = A \hat{x} - B K (x - e) \quad \text{eq.(4.36)}$$

A dinâmica do sistema global na forma de estado é obtida combinando a eq.(4.36) com o erro do observador. Portanto:

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - B K & B K \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} \hat{x} \\ e \end{bmatrix} \quad \text{eq.(4.37)}$$

A equação característica deste sistema de malha fechada é dada por:

$$\det \begin{bmatrix} sI - A + B K & -B K \\ 0 & sI - A + LC \end{bmatrix} = 0 \quad \text{eq.(4.38)}$$

Reescrevendo a eq.(4.38) temos:

$$\det(sI - A + B K) \cdot \det(sI - A + LC) = \alpha_c + \alpha_o = 0 \quad \text{eq.(4.39)}$$

Em outras palavras, o conjunto de pólos do sistema combinado, consiste na união dos pólos de controle com pólos do observador. Isto significa que o projeto da lei de controle e do observador podem ser independentemente executados, mesmo que estes sejam usados juntos, deste modo, os pólos permanecem inalterados.

A equação de estado do compensador é obtida incluindo a lei de realimentação ($u = -K\hat{x}$) na eq.(4.30) do observador. Portanto:

$$\dot{\hat{x}} = (A - B K - LC) \hat{x} + L y \quad \text{eq.(4.40)}$$

$$u = -K\hat{x} \quad \text{eq.(4.41)}$$

FIGURA 4.7 - Mecanismo do controlador e observador

4.4 INTRODUÇÃO DA ENTRADA DE REFERÊNCIA

O controlador obtido anteriormente é essencialmente o projeto de um regulador. Isto significa que as equações características do controle e do observador são escolhidas para rejeitar uma perturbação. Porém, esta abordagem não considera uma entrada de referência para alterar o ponto de ajuste do sistema.

Para solucionar este problema, as Figs. 4.8 e 4.9 mostram duas possibilidades para introduzir uma entrada de referência r no sistema:

FIGURA 4.8 - Compensador no caminho de realimentação da malha de controle

FIGURA 4.9 - Compensador no caminho direto da malha de controle

As figuras anteriores ilustram o assunto de uma forma geral, na qual a compensação pode ser realizada no caminho direto ou no caminho da realimentação da malha de controle. No entanto, a resposta do sistema para uma mesma referência de entrada é diferente, dependendo da configuração, porque os zeros da Função de Transferência são diferentes. Já os pólos de malha fechada são idênticos, porém, isto pode ser verificado facilmente fazendo $r = 0$ e notando que os sistemas são então idênticos.

A diferença nas respostas das duas configurações pode ser vista facilmente considerando o efeito de uma entrada degrau r .

Na Fig. 4.8, o degrau excitará o observador, justamente da mesma forma que ele excitará a planta, deste modo, o erro do observador permanecerá em zero durante e após o degrau. Isto significa que a dinâmica do observador não é excitada pelo comando de entrada, e por essa razão, a Função de Transferência de r para y tem zeros na mesma localização dos pólos do observador, na qual se cancelam. Como resultado, uma entrada degrau excitará o comportamento do sistema que é consistente somente com os pólos de controle, isto é, com as raízes de $\det(sI - A + BK) = 0$.

Na Fig. 4.9 uma entrada degrau r entra diretamente só no observador, deste modo causa um erro de observação que decai com as características dinâmicas do observador, além da resposta correspondente dos pólos de controle. Então, uma entrada degrau excitará o comportamento do sistema consistente com as raízes do controle e do observador, isto é, as raízes de $\det(sI - A + BK) \cdot \det(sI - A + LC) = 0$.

Por essa razão, a configuração mostrada na Fig. 4.8 é tipicamente superior na forma de comandar o sistema, FRANKLIN (1994).

4.4.1 ESTRUTURA PARA UMA ENTRADA DE REFERÊNCIA

Repetindo novamente as equações da planta e do controlador para o observador de estados de ordem plena, temos:

Planta:

$$\dot{\hat{x}} = A \hat{x} + B u \quad \text{eq.(4.42)}$$

$$\hat{y} = C \hat{x} \quad \text{eq.(4.43)}$$

Controlador:

$$\dot{\hat{x}} = (A - B K - L C) \hat{x} + L y \quad \text{eq.(4.44)}$$

$$u = -K \hat{x} \quad \text{eq.(4.45)}$$

Dada uma entrada de referência $r(t)$, a maneira linear mais geral para introduzir r nas equações do sistema é adicionar termos proporcionais na equação do controlador. Isto pode ser feito adicionando $N r$ na eq.(4.44) e $M r$ na eq.(4.45). Deve ser notado que neste caso, N é um escalar e M é um vetor $n \times 1$. Com estas adições as equações do controlador se tornam:

$$\dot{\hat{x}} = (A - B K - L C) \hat{x} + L y + M r \quad \text{eq.(4.46)}$$

$$u = -K \hat{x} + N r \quad \text{eq.(4.47)}$$

As alternativas mostradas nas Figs.(4.8 e 4.9) correspondem para diferentes escolhas de M e N . Uma vez que $r(t)$ é um sinal externo, isto é claro que nem M nem N afeta a equação característica do sistema combinado controlador-observador. Em termos de Função de Transferência, a seleção de M e N só afetará a transmissão de zeros de r para y e como consequência pode afetar significativamente a resposta transitória, mas não a estabilidade.

Existem três estratégias para escolher M e N para obter uma resposta transitória satisfatória:

a) Observador Autônomo: M e N são selecionados de forma que a equação do erro do observador de estados é independente de r . Ver Fig. 4.10.

b) Observador localizador de erro: M e N são selecionados de forma que somente o erro localizado, $e = (r - y)$ é usado no controle, Ver Fig. 4.11.

c) Observador com alocação de zeros: M e N são selecionados de forma que os n zeros da função transferência global são designados nos lugares de escolha do projetista. Ver Fig. 4.12.

4.4.1.1 OBSERVADOR AUTÔNOMO

Do ponto de vista de desempenho, este observador é bastante atrativo e o mais usado. Se $\hat{x}$ é para gerar uma boa estimativa de x , então seguramente o erro deverá ter uma excitação externa livre o quanto possível; isto é, o erro não deverá ser controlado proveniente de r .

A equação do erro do observador é obtida subtraindo a eq.(4.46) da eq.(4.44), com a saída da planta da eq.(4.43) substituída no observador da eq.(4.44) e o controle da eq.(4.45) substituído na planta da eq.(4.42):

$$\dot{\hat{x}} - \dot{\hat{x}} = A x + B(-K \hat{x} + Nr) - [(A - B K - LC) \hat{x} + Ly + Mr] \quad \text{eq.(4.48)}$$

$$\dot{e} = (A - LC) e + B N r - M r \quad \text{eq.(4.49)}$$

Se r não é para aparecer na eq.(4.49), então se deve escolher:

$$M = B N \quad \text{eq.(4.50)}$$

Uma vez que r é um escalar, e M é fixado para ser um fator constante. Logo, podemos escolher as equações de controlador como:

$$u = -K \hat{x} + Nr \quad \text{eq.(4.51)}$$

$$\dot{\hat{x}} = (A - LC) \hat{x} + B u + Ly \quad \text{eq.(4.52)}$$

Um diagrama de bloco correspondente às eqs.(4.51 e 4.52) é mostrado na Fig. 4.10 a seguir:

FIGURA 4.10 - Observador Autônomo

O problema deste esquema é que o controle é computado proveniente do ganho de realimentação e da entrada de referência, antes da mesma ser aplicada, e então o mesmo controle é entrada para a planta e o observador. Desta forma, se o controle da planta esta sujeito a uma saturação, o mesmo limite de controle será aplicado na eq.(4.52) para a entrada de controle da equação da estimativa $\hat{x}$. Este comportamento é essencial para o próprio desempenho do observador.

4.4.1.2 OBSERVADOR LOCALIZADOR DE ERRO

Esta técnica é usada para localizar o erro. Esta solução é às vezes forçada pelo projetista de controle quando o sensor medir somente o erro de saída. Por exemplo, em muitos termostatos a saída é a diferença entre a temperatura a ser controlada e o ponto de ajuste (*setpoint*) de temperatura, e não existe nenhuma indicação absoluta da temperatura de referência disponível para o controlador. Nesta situação, deve-se selecionar M e r de forma que a eq.(4.46) é somente governada pelo erro. Este requisito é satisfeito se selecionarmos $M=0$ e $N=-L$. Então a equação do observador é dada por:

$$\dot{\hat{x}} = (A - BK - LC) \hat{x} + L(y - r) \quad \text{eq.(4.53)}$$

O compensador neste caso, para projetos de baixa ordem, é um compensador no caminho direto de alimentação. Este projeto pode ter uma quantia considerável de *overshoot* devido ao zero no compensador. O diagrama de bloco correspondente para esta técnica é mostrado na Fig. 4.11.

FIGURA 4.11 - Observador localizador de erro

4.4.1.3 OBSERVADOR COM ALOCAÇÃO DE ZEROS

Este esquema determina M e N de forma que os valores dos zeros do sistema são escolhidos em localizações arbitrárias. Este método fornece a máxima flexibilidade em satisfazer a resposta transitória e a resposta em regime permanente. Os outros dois métodos são casos especiais deste terceiro método. Todos os três métodos dependem dos zeros.

Quando não existir nenhum observador e a entrada de referência for somada ao controle, os zeros de malha fechada do sistema permanecem fixos assim como os zeros da planta de malha aberta.

Quando o observador estiver presente e existir um zero de transmissão de r para u , então existirá necessariamente uma zero de transmissão de r para y , a menos que este haja um pólo na mesma localização do zero. É então suficiente tratar o controlador só para determinar qual o efeito das escolhas de M e R que levará a ter zeros no sistema. A equação para um zero de r para u proveniente das eqs.(4.46 e 4.47) é dada por:

$$\det \begin{bmatrix} sI - A + BK + LC & -M \\ -K & N \end{bmatrix} = 0 \quad \text{eq.(4.54)}$$

Onde os zeros do caminho direto são obtidos através da seguinte equação:

$$\det \left(sI - A + B K + L C - \frac{M}{N} K \right) = \gamma(s) = 0 \quad \text{eq.(4.55)}$$

A eq.(4.55) está exatamente na forma da eq.(4.33) para determinar L e produzir as localizações desejadas dos pólos para o observador. Aqui também temos que determinar M/N para um zero desejado na Função de Transferência da entrada de referência até o controle. Deste modo a seleção de M fornece uma liberdade para influenciar na resposta transitória. Também é permitido somar um polinômio arbitrário de ordem n na Função de Transferência de r para u e conseqüentemente de r para y ; isto é, pode-se projetar n zeros além de todos os pólos que foram projetados previamente.

Se as raízes de $\gamma(s)$ não são canceladas pelos pólos do sistema, então estes serão incluídos nos zeros de transmissão de r para y .

No entanto, duas considerações podem dirigir na escolha de M/N , isto é, na localização dos zeros. A primeira é a resposta dinâmica, em que os zeros influenciam significativamente na resposta transitória e a segunda é erro em regime permanente.

Um diagrama de bloco do observador com alocação de zeros é mostrado na Fig. 4.12

FIGURA 4.12 - Observador em alocação de zeros

Capítulo 5

Controle Adaptativo por Modelo de Referência

Neste capítulo apresenta-se vários esquemas de controle adaptativos, bem como o que será empregado nesta tese como bloco funcional.

5.1 INTRODUÇÃO

O funcionamento adequado dos processos industriais para alcançar os objetivos previstos depende muito da aplicação de estratégias de controle, na qual realizam operações eficientes e desta forma consegue eventuais mudanças nas condições de operação do sistema.

Em muitos casos, a presença de distúrbios e imprevistos variam amplamente os parâmetros do sistema e os sinais de entrada. Situações na qual muitas vezes um controlador convencional não pode manter o desempenho do sistema em níveis aceitáveis. Por essa razão, há a necessidade do emprego de um tipo especial de controle denominado controle adaptativo. Este pode sentir e corrigir automaticamente sempre que perturbações drásticas e/ou mudanças severas ocorram nos parâmetros do sistema, obtendo assim as características desejadas do projeto inicial.

A idéia de se utilizar controle adaptativo foi introduzida há aproximadamente quatro décadas atrás com a finalidade de se obter um controle efetivo de sistemas em ambientes desconhecidos. Um sistema que pudesse auto se ajustar, sem um grande

conhecimento do ambiente em que este operasse, era uma meta a ser alcançada naquela época.

Com o avanço da eletrônica dos dias atuais, a implementação de diversos algoritmos adaptativos tornou-se muito simples e de custo razoável devido à disponibilidade e aumento da versatilidade do *hardware* digital. Isto possibilitou resolver outros diversos tipos de problemas práticos, tais como: mudanças na função transferência da planta, perturbações estocásticas, comportamento não linear do sistema, mudanças na natureza da entrada e outros fatores.

O grande desenvolvimento na área tem sido observado, tanto nas universidades quanto nas indústrias. Diversos controladores adaptativos comerciais baseados em diferentes idéias estão aparecendo no mercado, e seu uso industrial tem crescido lentamente, mas de maneira estável e com confiabilidade (BRANCO, 1996).

Neste aspecto, este capítulo aborda os esquemas adaptativos mais comumente utilizados em controladores industriais e trata também do esquema utilizado para realizar o controle adaptativo através das redes *Fieldbus* em sistemas multivariáveis.

5.2 ESQUEMAS DE CONTROLE ADAPTATIVOS

Um sistema de controle adaptativo deve fornecer informações sobre o presente estado da planta, que é, identificar o processo, comparar o desempenho presente com o desempenho desejado, tomar a decisão para adaptar o sistema de forma a tender para o desempenho desejado, e finalmente iniciar a própria modificação de forma a dirigir o sistema para o desempenho desejado. Para isto os sistemas adaptativos apresentam uma malha de controle com realimentação, com o controlador possuindo parâmetros ajustáveis. O objetivo é encontrar uma maneira conveniente de se alterar os parâmetros do controlador em resposta às variações nas dinâmicas do processo e dos distúrbios. Para esta finalidade existem os esquemas de controle adaptativos. Sendo três, os principais que se destacam:

- (a) Ganho programado;
- (b) Controle com auto ajuste (*self-tuning*);
- (c) Controle por modelo de referência.

Tais esquemas diferem apenas no modo em que os parâmetros do controlador são ajustados (ÅSTRÖM & WITTENMARK, 1989).

5.2.1 GANHO PROGRAMADO

Em muitos casos é possível encontrar variáveis medidas que correlacionam bem com as mudanças nas características dos processos dinâmicos. Estas variáveis podem então ser usadas para mudar os parâmetros do controlador. Esta abordagem é chamada de "ganho programado" porque o esquema foi originalmente usado para medir o ganho e então mudar o controlador para compensar as mudanças no processo. Um diagrama de blocos do sistema com ganho programado é mostrado pela Fig. 5.1.

FIGURA 5.1- Diagrama de blocos de um sistema com ganho programado

O sistema pode ser visto como tendo duas malhas. Uma malha interna composta pelo processo e por um controlador, e uma outra malha externa que ajusta os parâmetros do controlador com base nas condições operantes. O ganho programado pode ser considerado como um mapeamento dos parâmetros do processo para atuar nos parâmetros do controlador. Isto pode ser implementado como uma função ou como uma tabela.

O ganho programado é uma compensação de malha aberta e pode ser visto como um sistema de controle por realimentação, na qual os ganhos são ajustados por uma compensação de pré-alimentação (*feedforward*). Não existe realimentação no desempenho do sistema de malha fechada, na qual compensará uma tabela incorreta.

O conceito de ganho programado originou-se do desenvolvimento de sistemas de controle de voo. Nesta aplicação, a pressão dinâmica é medida por sensores de dados aéreos e usada como uma variável programada. No controle de processos, a taxa de produção pode ser escolhida como uma variável programada, uma vez que constantes de tempo e atrasos são quase sempre inversamente proporcionais a esta taxa.

Embora esse método exija um bom conhecimento sobre o processo e que algumas variáveis auxiliares sejam medidas, este provém da vantagem que os parâmetros do controlador tem em se adaptar rapidamente as condições de mudanças na resposta do processo. O ganho programado também é um método muito útil para ser utilizado no controle de direção de navios, no controle de pH, no controle de combustão, no controle de motores e outros.

Historicamente, existe a controvérsia se o ganho programado poderia ser considerado como um sistema adaptativo ou não, porque os parâmetros variam em malha aberta. Contudo, o ganho programado é uma técnica muito útil para redução dos efeitos das variações dos parâmetros.

5.2.2 CONTROLE COM AUTO AJUSTE (*SELF-TUNING*)

A utilização de controladores com auto ajuste (*self-tuning*) é uma outra forma importante no controle adaptativo de sistemas. Estes são fáceis de implementar e são aplicados em processos complexos com uma ampla variedade de características envolvendo parâmetros desconhecidos, dinâmicas do processo variando com o tempo e perturbações estocásticas (CHALAM, 1987).

O procedimento básico de um controlador com auto ajuste é utilizar um projeto para conhecer os parâmetros da planta, e aplicá-los a planta desconhecida usando valores recursivamente estimados destes parâmetros. Um diagrama de blocos de um controle com auto ajuste é mostrado na Fig. 5.2.

FIGURA 5.2 - Sistema de controle adaptativo com auto ajuste

O controle também é realizado por duas malhas. A malha interna consiste do processo e do controlador. Os parâmetros do controlador são ajustados pela malha externa, que é composta por um observador recursivo e das especificações de projeto.

Para se obter uma boa estimação, às vezes pode ser necessário introduzir sinais de controle ou distúrbios. Um controlador com esta construção é dito auto ajustável para enfatizar que o controlador automaticamente ajusta seus parâmetros para obter as propriedades desejadas do sistema de malha fechada.

O bloco "Projeto do Controlador" representa uma solução *on-line* para o projeto de um sistema com parâmetros conhecidos. Isto é o "problema de projeto básico". Para análise de sistemas de controle adaptativo, torna-se útil a obtenção de tal projeto básico, porque este dá as características do sistema sob condições ideais, quando os parâmetros forem conhecidos (ÅSTRÖM & WITTENMARK, 1995).

O controlador com auto ajuste é um esquema muito flexível em relação à escolha dos métodos de projeto e estimação. Muitas combinações diferentes têm sido exploradas. Na Fig. 5.2 os parâmetros do controlador são ajustados indiretamente através de cálculos de projeto de um controlador com auto ajuste. Este controlador é denominado de controlador com auto ajuste explícito. Algumas vezes é possível reparametrizar o processo tal que ele possa ser implementado em termos dos parâmetros do controlador. Isto simplifica em demasia o algoritmo, porque os cálculos de projeto são eliminados. Este controlador é chamado de controlador com auto ajuste implícito, pois é baseado na estimação de um modelo implícito do processo.

5.2.3 CONTROLE ADAPTATIVO POR MODELO DE REFERÊNCIA (CAMR)

Controle adaptativo por modelo referência (CAMR) é um dos esquemas mais importantes do controle adaptativo. Este pode ser considerado como um servo sistema adaptativo na qual o desempenho desejado é expresso em termos de um modelo referência, na qual fornece a resposta desejada para um determinado sinal de comando. Um diagrama de blocos de um sistema de controle por modelo referência é mostrado na Fig. 5.3.

FIGURA 5.3 - Sistema de controle adaptativo por modelo de referência

O sistema tem uma malha de realimentação interna composta pelo processo e pelo controlador, e uma outra malha de realimentação externa que ajusta os parâmetros do controlador de maneira que o erro entre a saída do processo e a saída do modelo de referência torne-se mínimo. O mecanismo para ajustar os parâmetros no sistema adaptativo por modelo de referência pode ser obtido das seguintes formas:

- (a) Método do gradiente ou regra de MIT
- (b) Teoria da estabilidade de *Lyapunov*

5.2.3.1 MÉTODO DO GRADIENTE

Segundo ÅSTRÖM (1995), o método do gradiente foi a abordagem original do controle adaptativo por modelo de referência. Para apresentar o método do gradiente, será considerando um sistema de malha fechada na qual o controlador tem um parâmetro ajustável θ . A resposta de malha fechada é especificada por um modelo de referência cuja saída é y_m . O erro e é a diferença entre a saída da planta de malha fechada y_p e a saída do modelo de referência y_m , conforme é mostrado pela Fig. 5.4.

FIGURA 5.4 - Sistema de controle adaptativo pelo método do gradiente

Uma maneira de ajustar os parâmetros é minimizar a função dada pela eq.(5.1):

$$J(\theta) = \frac{1}{2} e^2 \quad (5.1)$$

Para tornar J pequeno, é necessário mudar os parâmetros na direção negativa do gradiente de J , de forma que:

$$\frac{\partial \theta}{\partial t} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (5.2)$$

Este é o denominado método do gradiente. A derivada parcial de $\frac{\partial e}{\partial \theta}$ é chamada de derivada sensível do sistema, dizendo como o erro é influenciado pelos parâmetros ajustados. Se for assumido que as mudanças nos parâmetros são mais lentas que mudanças em outras variáveis, então a derivada de $\frac{\partial e}{\partial \theta}$ pode ser avaliada sob a suposição que θ é constante. No entanto, existem muitas alternativas para a função dada pela eq.(5.1). Uma delas é escolher:

$$J(\theta) = |e| \quad (5.3)$$

Logo o método do gradiente torna-se:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial e}{\partial \theta} \text{sign } e \quad (5.4)$$

O primeiro CAMR foi implementado baseado nesta equação. Porém, existem muitas outras possibilidades, como por exemplo:

$$\frac{d\theta}{dt} = -\gamma \text{sign} \left(\frac{\partial e}{\partial \theta} \right) \text{sign}(e) \quad (5.5)$$

5.2.3.2 TEORIA DA ESTABILIDADE DE *LYAPUNOV*

A teoria da estabilidade de *Lyapunov* analisa o comportamento do sinal de uma função $V(x,t)$ e de sua derivada em relação ao tempo $\dot{V}(x,t) = dV(x,t)/dt$ para fornecer a informação a respeito da estabilidade ou da instabilidade de um estado de equilíbrio sem requerer diretamente a obtenção da solução (OGATA, 1997).

Pode se observar que se uma função escalar $V(x)$, onde x é um vetor n -dimensional, for positiva definida, então o estado x que satisfizer $V(x) = C_L$ em que C_L é uma constante positiva, permanece sobre uma hipersuperfície fechada do espaço de estados n -dimensional, ao menos nas vizinhanças da origem. Se $V(x) \rightarrow \infty$ à medida que $\|x\| \rightarrow \infty$, então esta superfície se estende sobre todo o espaço de estados. A hipersuperfície $V(x) = C_{L1}$ permanece inteiramente no interior da hipersuperfície $V(x) = C_{L2}$ se $C_{L1} < C_{L2}$.

Para um dado sistema, se for possível encontrar uma função escalar positiva definida $V(x)$ tal que sua derivada em relação ao tempo, ao longo de uma trajetória, seja sempre negativa, então, a medida que o tempo cresce, $V(x)$ toma valores de C_L cada vez menores. A medida que o tempo cresce, $V(x)$ se contrai a zero e, em consequência, x tende a zero. Isto implica na estabilidade assintótica da origem do espaço de estados.

O teorema da estabilidade de *Lyapunov*, que é uma generalização do fato precedente, propicia uma condição suficiente para a estabilidade assintótica. Este teorema pode ser enunciado da seguinte forma:

Suponha-se que um sistema seja descrito por $\dot{x} = f(x,t)$, onde $f(0,t) = 0$, qualquer que seja t . Se existir uma função escalar $V(x,t)$ que possua primeiras derivadas parciais contínuas e satisfaça as condições:

- a) $V(x,t)$ é definida positiva;
- b) $\dot{V}(x,t)$ é definida negativa.

Se, $V(x,t) \rightarrow \infty$ à medida que $\|x\| \rightarrow \infty$, então o estado de equilíbrio na origem é assintoticamente estável no sentido amplo, isto é, em grande escala (OGATA, 1997).

5.2.3.2.1 IDENTIFICAÇÃO DO PROBLEMA DE CONTROLE

Considere uma planta linear invariante no tempo de múltiplas entradas e múltiplas saídas com parâmetros desconhecidos. A planta possui ordem n^{th} cujos estados são acessíveis e descritos pela seguinte equação diferencial:

$$\dot{x}_p(t) = A_p x_p(t) + B_p u(t) \quad (5.6)$$

Onde A_p e B_p são matrizes desconhecidas, $A_p \in \mathbb{R}^{n \times n}$, $B_p \in \mathbb{R}^{n \times p}$, A_p é uma matriz assintoticamente estável e u é limitado. Um observador da forma:

$$\dot{\hat{x}}_p(t) = A_m \hat{x}_p(t) + [\hat{A}_p(t) - A_m] x_p(t) + \hat{B}_p(t) u(t) \quad (5.7)$$

É escolhido, onde $A_m \in \mathbb{R}^{n \times n}$ é uma matriz assintoticamente estável. $\hat{A}_p \in \mathbb{R}^{n \times n}$ e $\hat{B}_p \in \mathbb{R}^{n \times p}$ são matrizes cujos elementos podem ser ajustados pelo projetista. Se o erro dos estados e os erros dos parâmetros são definidos como:

$$e(t) \triangleq \hat{x}_p(t) - x_p(t) \quad (5.8)$$

$$\Phi(t) \triangleq \hat{A}_p(t) - A_p \quad (5.9)$$

$$\Psi(t) \triangleq \hat{B}_p(t) - B_p \quad (5.10)$$

Então a equação do erro é dada por:

$$\dot{e}(t) = A_m e(t) + \Phi(t) x_p(t) + \Psi(t) u(t) \quad (5.11)$$

O problema é ajustar os elementos das matrizes $\hat{A}_p(t)$ e $\hat{B}_p(t)$ ou o equivalente de $\Phi(t)$ e $\Psi(t)$ de forma que os valores de $e(t)$, $\Phi(t)$ e $\Psi(t)$ tendam a zero quando $t \rightarrow \infty$ (CHALAN, 1987). Portanto uma lei adaptativa é escolhida para ser:

$$\dot{\hat{A}}_p(t) = \dot{\Phi}(t) = -P e(t) x_p^T(t) \quad (5.12)$$

$$\dot{\hat{B}}_p(t) = \dot{\Psi}(t) = -P e(t) u^T(t) \quad (5.13)$$

Onde P é uma matriz $n \times n$ definida positiva simétrica ($P > 0$), na qual depende de A_m . É mostrado a seguir que estado de equilíbrio ($e(t) = 0$, $\Phi(t) = 0$ e $\Psi(t) = 0$) das eq.(5.11, 5.12 e 5.13) é uniformemente estável quando $\lim_{t \rightarrow \infty} e(t) = 0$.

Pela análise de estabilidade por *Lyapunov* é considerada $V(e, \Phi, \Psi)$ para ser a candidata a função de *Lyapunov*, da seguinte forma:

$$V(e, \Phi, \Psi) = e^T P e + \text{Tr}(\Phi^T \Phi + \Psi^T \Psi) \quad (5.14)$$

Onde $\text{Tr}(A_p)$ denota as características da matriz A_p . Se P é uma matriz definida positiva, então a função $V(e, \Phi, \Psi)$ é definida positiva. A derivada do tempo da eq.(5.14) ao longo da trajetória da eq.(5.11) é dada por:

$$\dot{V} = e^T (PA_m + A_m^T P)e + 2e^T P \Phi x_p + 2e^T P \Psi u + \text{Tr}(2\dot{\Phi}^T \Phi + 2\dot{\Psi}^T \Psi) \quad (5.15)$$

Uma vez que, A_m é uma matriz assintoticamente estável, a solução da equação matricial é dada por:

$$A_m^T P + PA_m = -Q < 0 \quad (5.16)$$

Onde Q é qualquer matriz $n \times n$ simétrica definida positiva, que produz uma matriz simétrica definida P . Com a lei adaptativa como definida nas eq.(5.12 e 5.13), temos:

$$\begin{aligned} \dot{V} = & -e^T Q_0 e + 2e^T P \Phi x_p - 2 \text{Tr}(x_p e^T P \Phi) + 2e^T P \Phi u - \\ & + 2 \text{Tr}(u e^T P \Psi) \end{aligned} \quad (5.17)$$

$$\dot{V} = -e^T Q_0 e \leq 0 \quad (5.18)$$

Uma vez que $\text{Tr}(ab^T) = b^T a$, onde a e b são vetores colunas. Sendo que o estado de equilíbrio ($e(t) = 0$, $\Phi(t) = 0$ e $\Psi(t) = 0$) das eq.(5.11, 5.12 e 5.13) é globalmente estável. Além disso o erro (e) é limitado, na qual implica em:

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (5.19)$$

5.2.3.2.2 O PROBLEMA DE CONTROLE

O problema de controle da planta linear, invariante no tempo, de ordem n^{th} com parâmetros desconhecidos e com variáveis de estado acessíveis pode ser descrito como segue. Uma planta para ser adaptativamente controlada pode ser descrita pela equação diferencial:

$$\dot{x}_p = A_p x_p + B_p u \quad (5.20)$$

Onde $A_p \in \mathbb{R}^{n \times n}$ e $B_p \in \mathbb{R}^{n \times n}$ são matrizes constantes e desconhecidas e o par (A_p, B_p) é controlável. O modelo referência é descrito pela equação diferencial:

$$\dot{x}_m = A_m x_m + B_m r \quad (5.21)$$

Onde, $A_m \in \mathbb{R}^{n \times n}$ é uma matriz assintoticamente estável, $B_m \in \mathbb{R}^{n \times n}$ e r uma entrada de referência limitada. É assumido que $x_m(t)$, para todo $t \geq 0$, representa uma trajetória desejada para $x_p(t)$ seguir. O objetivo é determinar um método para controlar a planta de forma que:

$$\lim_{t \rightarrow \infty} |x_p(t) - x_m(t)| = 0 \quad (5.22)$$

A solução para o problema de controle pode ser alcançada sob diferentes suposições relativas às informações disponíveis da planta. Serão considerados três casos:

- (a) Ajuste direto nos parâmetros da planta;
- (b) Controle realimentado com o parâmetro B_p conhecido;
- (c) Controle realimentado com o parâmetro B_p desconhecido.

5.2.3.2.2.1 AJUSTE DIRETO DOS PARÂMETROS DA PLANTA

É assumindo que os parâmetros da planta, na qual são os elementos A_p e B_p , são desconhecidos, e suas derivadas no tempo podem ser ajustadas usando os dados de entrada e saída. Se $u(t) = r(t)$ para $\forall t \geq t_0$, a equação diferencial da planta e do modelo podem ser descrita por:

$$\dot{x}_p = A_p(t) x_p + B_p(t) r \quad (5.23)$$

$$\dot{x}_m = A_m x_m + B_m r \quad (5.24)$$

Assim como visto anteriormente na identificação do problema, a lei adaptativa:

$$\dot{A}_p = P e x_p^T \quad (5.25)$$

$$\dot{B}_p = P e r^T \quad (5.26)$$

Resulta em $\lim_{t \rightarrow \infty} e(t) = 0$, onde $e(t) = x_p(t) - x_m(t)$, e a matriz simétrica definida positiva P é a solução da equação matricial:

$$A_m^T P + P A_m = -Q \quad (5.27)$$

Onde $Q > 0$. Este é o objetivo primário do controle, o fato do erro dos parâmetros poderem não tender a zero, não é diretamente relevante aqui.

Um diagrama de blocos de um sistema de controle adaptativo por modelo de referência com ajuste direto nos parâmetros da planta é mostrado pela Fig. 5.5.

FIGURA 5.5 - Sistema de controle adaptativo por modelo de referência com ajuste direto nos parâmetros da planta

5.2.3.2.2.2 CONTROLE COM REALIMENTAÇÃO COM B_p CONHECIDO

Considerando a equação diferencial da planta descrita por:

$$\dot{x}_p = A_p x_p + B_p u \quad (5.28)$$

Onde $A_p \in \mathbb{R}^{n \times n}$, $B_p \in \mathbb{R}^{n \times q}$ são matrizes constantes e desconhecidas e (A_p, B_p) é controlável. O objetivo do controle é escolher um vetor de entrada $u \in \mathbb{R}^q$ de maneira que todos os sinais na planta de malha fechada sejam limitados e os estados $x_p \in \mathbb{R}^n$ da planta sigam os estados $x_m \in \mathbb{R}^n$ do modelo de referência especificado pela seguinte equação diferencial:

$$\dot{x}_m = A_m x_m + B_m r \quad (5.29)$$

Onde $A_m \in \mathbb{R}^{n \times n}$ é uma matriz estável, $B_m \in \mathbb{R}^{n \times q}$, e $r \in \mathbb{R}^q$ é um vetor de referência de entrada limitado. O modelo de referência e a entrada $r \in \mathbb{R}^q$ são escolhidos de forma que $x_m(t)$ represente a trajetória desejada que x_p tem que seguir.

Se as matrizes A_p, B_p forem conhecidas, pode-se aplicar a lei de controle:

$$u = K^* x_p + L^* r \quad (5.30)$$

E obter a planta de malha fechada:

$$\dot{x}_p = (A_p + B_p K^*) x_p + B_p L^* r \quad (5.31)$$

Uma vez que, se $K^* \in \mathbb{R}^{q \times n}$ e $L^* \in \mathbb{R}^{q \times q}$ são escolhidos para satisfazer a equação algébrica:

$$A_p + B_p K^* = A_m \quad (5.32)$$

$$B_p L^* = B_m \quad (5.33)$$

Então a matriz de transferência de malha fechada da planta é mesma que a do modelo de referência e $x_p(t) \rightarrow x_m(t)$ exponencialmente para qualquer sinal de referência de entrada limitado $r(t)$. Deve se notar que dadas as matrizes A_p, B_p, A_m e B_m , nenhum K^* e L^* pode existir para satisfazer a condição de equilíbrio das eq.(5.32 e 5.33) indicando que a lei de controle da eq.(5.30) não pode ter uma flexibilidade bastante estrutural para encontrar o objetivo de controle. Em alguns casos, se a estrutura de A_p, B_p é conhecida, então A_m, B_m pode ser projetado de forma que as eq.(5.32 e 5.33) tenha uma solução para K^* e L^* .

Assumindo que K^* e L^* nas eq.(5.32 e 5.33) existam, isto é, que haja uma flexibilidade estrutural suficiente para encontrar o objetivo de controle, e proposta a lei de controle:

$$u = K(t) x_p + L(t) r \quad (5.34)$$

Onde $K(t)$ e $L(t)$ são estimativas K^* e L^* , respectivamente, para ser gerada uma apropriada lei adaptativa.

Adicionando e subtraindo o termo da entrada desejada, normalmente $B_p(K^* x_p - L^* r)$, na equação da planta e usando as eq.(5.32 e 5.33), obtêm-se:

$$\dot{x}_p = A_m x_p + B_m r + B_p(-K^* x_p - L^* r + u) \quad (5.35)$$

Na qual pode ser mostrado que o erro entre o modelo e a planta ($e = x_p - x_m$) e erro dos parâmetros ($\tilde{K} \triangleq K - K^*$, $\tilde{L} \triangleq L - L^*$) satisfazem a equação:

$$\dot{e} = A_m e + B_p(\tilde{K} x_p + \tilde{L} r) \quad (5.36)$$

Na qual depende da matriz desconhecida B_p . Assumindo que o sinal de B_p é conhecido, então L^* é definido positivo ou negativo e $\Gamma^{-1} = L^* \operatorname{sgn}(\ell)$, onde $\ell = 1$ se L^* é definido positivo e $\ell = -1$ se L^* é definido negativo. Logo $B_p = B_m L^{*-1}$ e a eq.(5.36) torna-se:

$$\dot{e} = A_m e + B_m L^{*-1}(\tilde{K} x_p + \tilde{L} r) \quad (5.37)$$

Propondo a seguinte função a candidata de *Lyapunov*:

$$V(e, \tilde{K}, \tilde{L}) = e^T P e + \operatorname{Tr}(\tilde{K}^T \Gamma^{-1} \tilde{K} + \tilde{L}^T \Gamma^{-1} \tilde{L}) \quad (5.38)$$

Onde $P = P^T > 0$ satisfaz a equação de *Lyapunov*:

$$P A_m + A_m^T P = -Q \quad (5.39)$$

Para o mesmo $Q = Q^T > 0$. Então:

$$\dot{V} = -e^T Q e + 2e^T P B_m L^{*-1} (\tilde{K} x_p + \tilde{L} r) + 2 \text{Tr} (\tilde{K}^T \Gamma \dot{\tilde{K}} + \tilde{L}^T \Gamma \dot{\tilde{L}}) \quad (5.40)$$

Agora:

$$e^T P B_m L^{*-1} \tilde{K} x_p = \text{Tr} (x_p^T \tilde{K}^T \Gamma B_m^T P e) \text{sgn}(\ell) = \text{Tr} (\tilde{K}^T \Gamma B_m^T P e x_p^T) \text{sgn}(\ell) \quad (5.41)$$

e

$$e^T P B_m L^{*-1} \tilde{L} r = \text{Tr} (\tilde{L}^T \Gamma B_m^T P e r^T) \text{sgn}(\ell) \quad (5.42)$$

Por essa razão:

$$\dot{\tilde{K}} = \dot{K} = -B_m^T P e x_p^T \text{sgn}(\ell) \quad (5.43)$$

$$\dot{\tilde{L}} = \dot{L} = -B_m^T P e r^T \text{sgn}(\ell) \quad (5.44)$$

Temos

$$\dot{V} = -e^T Q e \quad (5.45)$$

O esquema de controle adaptativo desenvolvido é dado pelas eq.(5.34) e eq.(5.43 e 5.44). A matriz $B_m P$ atua como uma matriz de ganho adaptativo, onde P é obtido resolvendo a equação de *Lyapunov*: $P A_m + A_m^T P = -Q$, para algumas matrizes arbitrárias $Q = Q^T > 0$. Diferentes escolhas da matriz Q não afetaram o comportamento assintótico do esquema, mas estas afetarão a resposta transitória. A suposição que o desconhecido L^* no equilíbrio da equação $B L^* = B_m$ é definido positivo ou negativo e impõe uma restrição adicional na estrutura e elementos de B , B_m . Porque B é desconhecido e esta suposição não pode ser realista em algumas aplicações.

Um diagrama de blocos de um sistema de controle adaptativo por modelo de referência com ajuste dos parâmetros do controlador é mostrado na Fig. 5.6.

FIGURA 5.6 - Diagrama de blocos de um sistema de controle por modelo de referência com ajuste nos parâmetros do controlador

5.2.3.2.2.3 CONTROLE COM REALIMENTAÇÃO B_p DESCONHECIDO

Quando B_p é desconhecido o problema torna-se consideravelmente mais complexo. Uma vez que não é mais possível escolher uma matriz B_m do modelo de referência de forma que $B_m = B_p$, logo o projeto do controlador deve ser baseado em suposições adicionais da estrutura de B_p . Em particular, é assumido que matrizes constantes de K^* e L^* existem tal que:

$$A_p + B_p K^* = A_m \quad (5.46)$$

$$B_p L^* = B_m \quad (5.47)$$

As estruturas de dois controladores são mostradas nas Fig. 5.7 e 5.8.

FIGURA 5.7 - Estrutura do controlador com equações de erro não-lineares no ajuste dos parâmetros

FIGURA 5.8 - Estrutura do controlador com ajuste de parâmetros usando sinais do sistema

Uma análise do sistema adaptativo da Fig. 5.7 resulta em equações de erro que são não-lineares nos erros dos parâmetros, de forma que os métodos descritos deste modo não podem ser usados para gerar uma lei adaptativa estável (NARENDRA & ANNASWAMY, 1989). Conseqüentemente, toda atenção é voltada para a estrutura do controlador mostrada como na Fig. 5.8. Os parâmetros da matriz ganho de pré-alimentação $L(t)$ e da matriz ganho de realimentação $K(t)$ são ajustados usando os sinais disponíveis do sistema. Se $L(t) \equiv L^*$ e $K(t) \equiv K^*$, isto segue que a matriz transferência da planta junto com o controlador é idêntica a do modelo.

Como mostrado a seguir, as leis adaptativas deduzidas para este problema asseguram a convergência dos parâmetros para zero somente quando valores iniciais dos parâmetros $L(t_0)$ e $K(t_0)$ são escolhidos próximos dos valores de L^* e K^* .

Quando a estrutura do controlador da Fig. 5.8 é usada, a entrada de controle

$u(t)$ pode ser expressa como:

$$u(t) = L(t) K(t) x_p(t) + L(t) r(t) \quad (5.48)$$

Isto produz um sistema de malha fechada descrito por:

$$\dot{x}_p = [A_p + B_p L(t) K(t)] x_p + B_p L(t) r \quad (5.49)$$

Proveniente das eq.(5.49) e (5.29), obtêm-se a equação do erro:

$$\begin{aligned} \dot{e} &= A_m e + [A_p + B_p (L^* + L(t) - L^*) K(t) - A_m] x_p + [B_p (L(t) - L^*)] r \\ \dot{e} &= A_m e + B_m [K(t) - K^*] x_p + B_m [L^{*-1} L(t) - I] K(t) x_p + B_m [L^{*-1} L(t) - I] r \\ \dot{e} &= A_m e + B_m \tilde{K}(t) x_p + B_m \tilde{L}(t) u \end{aligned} \quad (5.50)$$

Onde: $\tilde{K}(t) \triangleq K(t) - K^*$ e $\tilde{L}(t) \triangleq L^{*-1} - L^{-1}(t)$.

Agora escolhe-se a lei adaptativa para ajustar os parâmetros $\tilde{K}(t)$ e $\tilde{L}(t)$, como:

$$\dot{\tilde{K}} = -B_m^T P e x_p^T \quad (5.51)$$

$$\dot{\tilde{L}} = -B_m^T P e u^T \quad (5.52)$$

Onde $P A_m + A_m^T P = -Q < 0$. Isto assegura a solução das eq.(5.50, 5.51 e 5.52), na qual pode ser mostrada a seguir, onde a função escalar:

$$V = e^T P e + \text{Tr}(\tilde{K}^T \tilde{K} + \tilde{L}^T \tilde{L}) \quad (5.53)$$

Tem a derivada do tempo ao longo da trajetória das eq.(5.50, 5.51 e 5.52) dada por:

$$\begin{aligned}\dot{V} &= -e^T Q e + 2e^T P B_m \tilde{K} x_p + 2e^T P B_m \tilde{L} u - \text{Tr}(2x_p e^T P B_m \tilde{K}) \\ &\quad - \text{Tr}(2u e^T P B_m \tilde{L}) \\ \dot{V} &= -e^T Q e < 0\end{aligned}\quad (5.54)$$

O controlador adaptativo pode ser implementado em termos de $\dot{K}$ e $\dot{L}$ como:

$$\dot{K} = -B_m^T P e x_p^T \quad (5.55)$$

$$\dot{L} = -L B_m^T P e u^T L \quad (5.56)$$

Um diagrama de blocos de um sistema de controle adaptativo por modelo de referência com ajuste dos parâmetros do controlador com B_p desconhecido é mostrado na Fig. 5.9

FIGURA 5.9 - Diagrama de blocos de um sistema de controle por modelo de referência com ajuste nos parâmetros do controlador com B_p desconhecido

Este esquema de controle adaptativo por modelo de referência será projetado como um bloco funcional dentro dos padrões da norma *Fieldbus Foundation* com o objetivo de facilitar e melhorar o controle em plantas com múltiplas entradas e múltiplas saídas, trazendo desta forma uma contribuição na área do desenvolvimento de controle de processos contínuos com processamento distribuído em rede.

Capítulo 6

Projeto dos Novos blocos Funcionais

Neste capítulo apresenta-se o projeto dos novos blocos funcionais desenvolvidos de acordo com a norma *Fieldbus Foundation*.

6.1 BLOCOS FUNCIONAIS

6.1.1 INTRODUÇÃO

Os blocos funcionais representam funções básicas de automação como: aquisição, controle, atuação, monitoração e alarmes desempenhadas nos dispositivos de campo, após a definição da aplicação pelo usuário. Estes são modelos de aplicação e descrevem a informação para interface de operação, para os diagnósticos e a configuração destas funções (conexões: entradas saídas e parâmetros internos).

Cada bloco funcional processa parâmetros de entrada de acordo com um algoritmo específico e um conjunto interno de parâmetros, produzindo parâmetros que sejam válidos para uso no mesmo bloco funcional da aplicação ou de outro bloco funcional da aplicação (FIELDDBUS FOUNDATION, 1999).

Na configuração da estratégia de controle é especificada a escolha do bloco funcional adequado a ser utilizado e em qual dispositivo este será executado. A interligação dos blocos funcionais define a estratégia de controle e a programação do processo a ser controlado.

6.1.2 AS NORMAS

A norma *Fieldbus Foundation* com relação aos blocos funcionais se divide em cinco partes:

Na primeira parte é definida a arquitetura e a estrutura geral dos blocos e do respectivo processo de aplicação, sendo denominada parte da camada do usuário (FIELDBUS FOUNDATION - 890, 1999).

Na segunda parte são definidos dez blocos funcionais básicos que englobam os algoritmos fundamentais de controle, além dos blocos de entrada e saída. Também são definidos dois tipos particulares de blocos: o *resource block* e o *Transducer block* (FIELDBUS FOUNDATION - 891, 1999).

Na terceira parte são definidos onze blocos funcionais avançados garantindo assim a *Foundation* uma grande flexibilidade e abrangências quanto à implementação de diferentes malhas de controle distribuídas nos dispositivos de campo (FIELDBUS FOUNDATION - 892, 1999).

Na quarta parte define-se os blocos de múltiplas entradas e saídas, dedicados ao uso em Controladores Lógicos Programáveis (CLPs). Com estes blocos, variáveis analógicas ou digitais do sistema *Fieldbus* podem ser utilizadas pelos controladores lógicos programáveis e vice-versa (FIELDBUS FOUNDATION-893, 1999).

Na quinta parte é definido o *Flexible Function Block* para uso em CLPs. Sua estrutura e algoritmo são projetados pelo usuário no próprio local do uso (FIELDBUS FOUNDATION - 894, 1999).

As normas acima devem ser seguidas na implementação dos blocos funcionais definidos. No entanto, a *Foundation* permite que os fabricantes possam adicionar novas funcionalidades aos blocos padrões e implementar novos blocos. Para isto é necessário que este processo seja realizado de modo que as novas implementações estejam de acordo com a norma para que os blocos possam ser integrados a um sistema *Foundation* de forma clara e transparente.

6.1.3 PROCESSO DE APLICAÇÃO DOS BLOCOS FUNCIONAIS

6.1.3.1 INTRODUÇÃO

De acordo com a *Fieldbus Foundation*, as especificações do Processo de aplicação dos blocos funcionais (FBAP) visam normalizar:

- 1) O projeto e a implementação de aplicações de blocos funcionais de modo interoperável;
- 2) A configuração e a implementação de blocos funcionais em diferentes tipos de transmissores de campo;
- 3) A interoperação das aplicações dos blocos funcionais durante a operação do sistema.

A FBAP possui para tal, estruturas e definições que formam a base para a operação do sistema e para o seu entendimento (FIELDBUS FOUNDATION, 1998). Estas estruturas e definições são mostradas na Fig. 6.1.

FIGURA 6.1 - Componentes do FBAP

6.1.3.2 *RESOURCE*

Um *resource* é considerado uma subdivisão lógica do *software* ou do *hardware* do transmissor. Este tem o controle independente da sua própria operação, assim, é independente e pode ser modificado sem prejuízo de outras estruturas. Os processamentos, se utilizados, são feitos baseados em dados ou processos externos, acessados via estruturas de interfaceamento.

6.1.3.3 *FUNCTIONS BLOCKS* (BLOCOS FUNCIONAIS)

O bloco funcional é uma função de automação básica realizada por uma aplicação, sendo o mais independente possível do meio físico e da estrutura da rede. Os blocos funcionais processam parâmetros de entrada em função da sua configuração e assim geram saídas que serão utilizadas por outros blocos.

6.1.3.4 *TRANSDUCER BLOCK* (BLOCO TRANSDUTOR - TRD)

Os *transducer block* isolam os blocos funcionais dos dispositivos específicos de I/O, como sensores, atuadores e chaves. Os *transducers* controlam os dispositivos de I/O e geram uma interface padrão para os blocos funcionais, realizam também funções como calibração, filtragem de sinais e conversão de dados.

6.1.3.5 *LINK OBJECT* (OBJETO LINK)

Os *link objects* promovem a troca de dados entre os *resources* e a rede de comunicação. Todos os dados ou eventos que são enviados na rede devem ser definidos por *link objects*. Estas estruturas geram *trend reports*, e notificações de alerta.

6.1.3.6 *RESOURCE BLOCK* (BLOCO RESOURCE – RES)

As características específicas do *hardware* que são associadas a qualquer *resource* são mapeadas e se tornam visíveis através do *resource block*. Assim como o *transducer*, o *resource* isola características do hardware além de conter algumas funções relacionadas ao hardware.

6.1.3.7 *VIEW E TREND OBJECTS (OBJETOS VIEW E TREND)*

Estes objetos tomam o acesso a parâmetros dos blocos mais eficientes. O *view object* permite o pedido de um conjunto de parâmetros com uma só requisição e o *trend object* permite o acesso a um conjunto de amostras de um mesmo parâmetro de uma só vez.

6.1.3.8 *ALERT OBJECTS (OBJETOS ALERTA)*

Permite a notificação de eventos entre transmissores. Isto torna possível iniciar um procedimento de alarme em caso de um evento significativo.

6.1.3.9 *DOMAIN E PROGRAM INVOCATION OBJECTS (OBJETOS DOMÍNIO E PROGRAMA INVOCATION)*

Programas ou códigos podem ser definidos como *domain objects*, assim serviços de *download* ou *upload* tornam-se possíveis. O objeto *program invocation* pode associar a um *domain* um código executável.

6.1.3.10 *ACTION OBJECTS (OBJETOS DE AÇÃO)*

Um *resource* pode ou não conter um *action object*. Este objeto apaga um objeto ou um bloco dentro do *resource*.

6.1.3.11 *SYSTEM TIME (TEMPO DO SISTEMA)*

É uma sincronização gerada pelo *System Management* para o FBAP, para sincronizar a operação dos blocos funcionais. O *System Management* mantém um padrão de tempo comum a todas as aplicações da rede e todos os registros de tempo dos alarmes, eventos e *trends* são baseados nesta sincronização.

6.1.3.12 *FUNCTION BLOCK SCHEDULE* (ESCALONAMENTO DOS BLOCOS FUNCIONAIS)

O *System Management Kernel* estrutura um agendamento, ou *Function Block Schedule*, para a execução dos blocos de cada transmissor da rede, este agendamento está baseado no *system time*. A execução de um bloco pode ser também requisitada pela conclusão de outro bloco. Os métodos de montagem do sequenciamento dos blocos podem ser definidos pelo fabricante.

6.1.3.13 *NETWORK TRAFFIC* (TRÁFEGO DE REDE)

É todo o tráfego de informações do *function block application* via rede. Estas informações podem ser agendas ou não. Através das comunicações agendadas, os blocos enviam e recebem parâmetros sincronizadamente para que possam executar suas funções dentro do *schedule*. As mensagens não agendadas ou *background traffic* não interferem nas comunicações agendadas que são críticas.

6.1.4 CARACTERÍSTICAS DOS BLOCOS FUNCIONAIS NOS SISTEMAS DE CONTROLE DISTRIBUÍDOS EM REDES

Estes componentes apresentados anteriormente realizam funções, procedimentos e interações durante todo o processo de operação do sistema. Basicamente existem 5 tipos primários de operações características no sistema *Foundation Fieldbus*, e em geral, nos sistemas de controle distribuído. São estes:

6.1.4.1 CONFIGURAÇÃO DOS BLOCOS FUNCIONAIS

A identificação dos blocos funcionais é dada por um *tag* e por um índice no dicionário de objetos (OD). O *tag* é uma identificação simbólica do bloco, única em todo o sistema e o índice no OD é um número associado a bloco que otimiza a rotina de acesso ao bloco. O índice no OD ao contrario do *tag* só tem validade no recurso a qual o bloco está inserido.

Os parâmetros são visíveis e acessíveis pela rede, sendo que os parâmetros de entrada e de saída são usados para a troca de dados entre os blocos através dos *links* e os parâmetros de configuração apenas para operações dentro do bloco, portanto não podem ser “*linkados*”.

Todos os parâmetros estão associados a números de identificação no OD denominados *DD member Ids*, estes *Ids* são relativos apenas ao parâmetro, assim, a informação do bloco é necessária para diferenciar parâmetros semelhantes em blocos distintos.

A descrição de cada parâmetros de cada bloco do transmissor está presente também na respectiva DD.

6.1.4.2 CONFIGURAÇÃO DOS *LINKS* ENTRE OS BLOCOS FUNCIONAIS

Os parâmetros dos blocos necessitam ser acessados tanto por outros blocos, como por elementos de interface como configuradores e gerenciadores de *status* (ferramentas de diagnóstico). Cada elemento de comunicação é portanto configurado e associado a um objeto de *link*.

A comunicação entre dois blocos se dá por um *link* (local ou remoto) entre um parâmetro de saída de um bloco e um parâmetro de entrada de outro bloco.

A transferência de dados em um *link* remoto, ou seja entre transmissores, se dá através das camadas de rede por um canal de comunicação virtual conhecido denominado VCR (*virtual communication relationship*), como já apresentado anteriormente. A VCR utilizada em um *link* remoto deve ser identificada por um número (*VCR number*) e é configurada ou pré-determinada pelo fabricante.

Para suprir as necessidades do FBAP, dois tipos de VCR são utilizados: Cliente/Servidor para comunicações entre aplicações e *Publisher/Subscriber* onde uma aplicação envia um dado na rede para uma ou mais aplicações.

6.1.4.3 PROCESSAMENTO DOS ALARMES

Os alarmes e eventos, ou simplesmente alertas são mudanças nos estados do FBAP. Cada aplicação possui um objeto responsável por notificar os outro FBAP sobre a ocorrência de um alerta, estes elementos são os *alert objects*.

Os objetos de alerta executam todo o procedimento necessário em uma situação de alerta, aliviando assim os blocos *transducers*, *resource* e até mesmo o bloco funcional de procedimentos de emergência, permitindo que estes continuem nas suas respectivas funções usuais.

Os objetos de alerta monitoram constantemente a operação dos blocos, assim que seja identificada uma situação previamente definida como de alerta, ou algum alarme, o objeto publica na rede uma notificação do evento ocorrido associado a um registro de tempo. Uma confirmação é então esperado por parte do *link master* indicando que as devidas providências por parte da interface operacional já foram tomadas. caso esta confirmação não seja recebida, a mensagem de alerta é novamente publicada na linha.

6.1.4.4 ACESSO A INFORMAÇÕES PARA A INTERFACE DO OPERADOR, DIAGNÓSTICO E CONFIGURAÇÃO

Existem dois tipos de comunicação relativa à operação dos blocos funcionais as comunicações cíclicas e as acíclicas. As comunicações acíclicas são de baixa prioridade, para que não interfiram nas comunicações cíclicas e são utilizadas para informações de configuração, interface com operador e diagnóstico do sistema. Os parâmetros de um bloco são agrupados em quatro conjuntos para otimizar este tipo de comunicação acíclica, facilitando o acesso aos parâmetros por parte do configurador. Estes conjuntos são denominados *views*. Das quatro *views*, duas são dinâmicas e duas estáticas. A definição das *views* é de responsabilidade do usuário.

Outro tipo de estrutura de dados para otimização de comunicação acíclica é o *trend object*, onde são enviados 16 registros consecutivos de uma mesma variável.

6.1.4.5 COORDENAÇÃO DO FUNCIONAMENTO DOS BLOCOS E SUA OPERAÇÃO

A execução de um bloco é sempre disparada por um *input event*. A geração deste evento depende da configuração do sistema, e as fonte de *input event* são as seguintes:

1. *Schedule*, a taxa com que um bloco é executado é definido por um *schedule*, este *schedule* determina o instante em que o bloco inicia sua execução.
2. Final da execução de outro bloco, o final da execução de um bloco pode disparar outro desde que estes estejam no mesmo *resource*.
3. Especificação do fabricante, o fabricante pode especificar o evento de disparo do bloco.

O escalonamento da execução dos blocos de uma aplicação é influenciado pelas seguintes características de um sistema de controle distribuído:

- Todos os blocos devem compartilhar um senso comum de tempo, denominado tempo do sistema
- As aquisições devem ser realizadas em um período preciso para que possam ser disponibilizadas aos blocos de entrada e então aos parâmetros de entradas dos blocos de controle.
- Os parâmetros de entrada e de configuração devem ser atualizados corretamente de modo a garantir a presença de dados corretos durante a execução dos blocos.
- O período entre a geração de um parâmetro de saída e o início da execução de um bloco subsequente deve ser muito pequeno comparado ao período de execução dos blocos. Este período é composto pelo tempo de transferência do parâmetro e o tempo que o bloco leva para começar a iniciar após o parâmetro estar pronto.

Estas características tornam necessário o controle de três aspectos na execução dos blocos:

1. Deve ser possível sincronizar os blocos utilizando um único sistema de referência temporal em todo o sistema,
2. Deve ser possível gerar um *schedule* que garanta que os blocos gerem saídas antes do início da execução de outros blocos que utilizem estas saídas,
3. O *schedule* deve permitir o tráfego de outras mensagens independentemente da publicação escalonada de parâmetros

O *System Management Agent* realiza este controle de sincronização dos blocos e sincroniza o senso comum de tempo nos transmissores para todas as aplicações, atendendo assim todos os requisitos acima apresentados.

6.1.5 INTERAÇÕES EXTERNAS

Todos os dispositivos que contem blocos funcionais são chamados dispositivos de campo. Existem 3 outros tipos de dispositivos que podem estar conectados a uma rede, e não que possuem blocos funcionais:

1. Dispositivos temporários: são associados a um conjunto de endereços temporários, a aplicação presente neste tipo de equipamento geralmente é capaz de realizar tarefas de diagnóstico e ajuste em outros dispositivos.
2. Dispositivos de Interface: possuem endereço definitivo no barramento, são utilizados como interface homem-máquina e dispositivos de configuração e diagnóstico da aplicação.
3. Dispositivos Monitores: não possuem endereço e são utilizados para monitorar a comunicação.

6.1.5.1 INTERAÇÃO COM APLICAÇÕES DE CONFIGURAÇÃO

Uma aplicação de configuração, ou configurador, pode estar em um dispositivo temporário ou de interface e é utilizada para modificar os blocos existentes assim como para apagar blocos.

Quando o configurador não tem informações sobre a disposição dos blocos, e portanto sobre o acesso aos parâmetros, este deve utilizar funções da *System Management* para determinar onde os blocos estão instanciados e qual VCR deve ser utilizada para acessar cada parâmetro. A partir daí são utilizados serviços da FMS para estabelecer as respectivas VCRs.

Para obter as informações completas dos parâmetros a serem acessados, o configurador utiliza os registros do DD, e assim a FMS pode ler e modificar os parâmetros remotamente de maneira adequada.

6.1.5.2 INTERAÇÃO COM INTERFACES HOMEM-MÁQUINA

A FMS utiliza números de identificação associados aos recursos do sistema, por motivo de otimização. Caso a Interface Homem-Máquina (IHM) não possua registros destes identificadores, ou então o endereço do recurso, deve utilizar serviços de procura de *tag* da FMS para obtê-los.

Quando um alerta é gerado, o recurso associado a este alerta utiliza serviços de notificação de eventos da FMS para acionar a IHM ativa.

6.1.5.3 INTERAÇÃO COM OUTRAS APLICAÇÕES DE CONFIGURAÇÃO

Na rede podem existir dispositivos tais como estações *host*, controladores SCD, CLPs ou outros elementos que não possuem blocos funcionais mas possuem aplicações que executam tarefas de controle. Este tipo de aplicação geralmente necessita acessar ou modificar parâmetros em blocos funcionais. Para tais ações, devem ser comunicados o parâmetro e o seu *status*.

O operador pode configurar o modo do bloco de modo a aceitar tais ações de controle remotas, assim certos parâmetros podem ser atualizados periodicamente por tais aplicações.

Caso o parâmetro a ser recebido não foi enviado dentro do período de tempo esperado, ou então foi recebido com status *bad*, o bloco muda de modo de operação para que este parâmetro possa ser atualizado por outro bloco ou pelo usuário, e assim que a comunicação com o sistema original for restabelecida, o bloco volta ao modo de operação anterior.

6.1.6 INTERAÇÃO COM OUTROS DISPOSITIVOS DE CAMPO

Um parâmetro de saída de um bloco pode ser ligado a um ou mais parâmetros de entrada em outros blocos, juntamente com a informação é transmitido seu *status*. Estes parâmetros estão sempre associados a uma unidade de engenharia. Caso um parâmetro não seja recebido, entende-se que ocorreu uma falha de comunicação e esta informação será utilizada pelo processamento de modo a evitar descontinuidades nas ações de controle.

Existem dois sentidos de fluxo de dados entre blocos funcionais:

- Fluxo de dados para frente: quando os dados vão de um bloco de entrada para um de controle e deste para um de saída.
- Fluxo de dados para trás: quando os dados vão de um bloco de saída para um bloco de controle e deste para um outro bloco de controle. O *status* desta informação diz se a informação foi possível ou não de ser enviada pelo bloco posterior.

6.1.7 BLOCOS FUNCIONAIS NO PROTOCOLO *FIELDBUS FOUNDATION*

A norma *Fieldbus Foundation* relativa ao processo de aplicação de blocos funcionais propõe um conjunto inicial de dez blocos padrões projetados para realizarem as tarefas básicas de aquisição, atuação e controle presentes em uma malha de controle clássica nos processos industriais. Estes blocos são:

1) *Analog Input* (AI) – Este bloco coleta os dados de entrada do bloco *transducer*, selecionado pelo número do canal, e torna-os disponíveis na sua saída para outros blocos funcionais. Este possui conversão de escala, filtragem e raiz quadrada.

2) *Discret Input* (DI) - Este bloco coleta dados discretos de entrada do bloco *transducer*, selecionado pelo número do canal, e torna-os disponíveis na sua saída para outros blocos funcionais.

3) *Manual Loader* (ML) – Este bloco tem a sua saída cujo valor não é estabelecida pelo algoritmo do bloco. Sua saída (OUT) pode ser estabelecida pelo operador no modo manual (Man) ou pelo computador no modo remoto (ROut).

4) *Bias/Gain Station* (BG) – Este bloco foi planejado para permitir que algumas unidades controladoras, como mestres de caldeiras, sejam colocadas como controladoras de saída. Este bloco também pode ser usado para um parcial controle externo de ação à frente (*feedforward*).

5) *Control Selector* (CS) – Este bloco foi planejado para selecionar um, dois ou três sinais de controle de maneira determinada pelo SEL_TYPE, quando o bloco está no modo automático (Auto).

6) *PD Controller* (PD) – Este bloco é exatamente parecido com o bloco PID descrito a seguir, com exceção de ter BIAS ao invés de RESET. O bloco PD é usado quando o próprio processo faz a integração.

7) *PID Controller* (PID) – Este bloco é utilizado na maioria dos esquemas de controle, e é quase usado universalmente. No algoritmo do PID, o ganho (GAIN) é aplicado para todos os termos do PID, o Proporcional e o Integral atuam sobre o erro e o Derivativo atua sobre o valor da variável do processo.

8) *Ratio Station* (RA) – O *setpoint* deste bloco é razão de sua saída pela sua entrada. A razão do *setpoint* de 0.5 produz uma saída que é a metade de sua entrada.

9) *Analog Output* (AO) – É um bloco funcional usado por dispositivos que trabalham com elementos de saída na malha de controle, como válvulas, atuadores e posicionadores. O bloco AO recebe um sinal de um outro bloco funcional e passa esse resultado para uma saída do bloco *transducer* através de um canal de referência interno.

10) *Discrete Output* (DO) - Este bloco é semelhante ao bloco AO, no entanto este trabalha com variáveis discretas.

Posteriormente, foram definidos mais onze blocos funcionais avançados que realizam tarefas específicas de cálculo e de controle. Estes Blocos são:

1) *Device Control* (DC) – Este bloco é designado para permitir o controle do ponto de ajuste (*setpoint*) de dispositivos discretos com múltiplos estados.

2) *Output Splitter* (OS) – Este bloco provém a capacidade de direcionar duas saídas de controle para uma única entrada.

3) *Signal Characterizer* (SC) – Este bloco tem duas seções, cada uma com uma saída que é uma função não-linear das respectivas entradas. A função é determinada por uma única tabela de coordenadas x-y de vinte pontos cada.

4) *Lead Lag* (LL) – Este bloco fornece uma compensação dinâmica para o valor do parâmetro de entrada (IN). O bloco pode funcionar como dispositivo *Lead*, *Lag* ou ambos.

5) *Deadtime* (DT) - Este bloco tem a capacidade de atrasar um sinal contínuo através de um valor de tempo especificado. O bloco recebe uma entrada e produz uma saída atrasada de acordo com o valor de tempo especificado pelo parâmetro *DEAD_TIME*.

6) *Integrator (IT)*– Este bloco integra uma variável como função do tempo ou através de acumuladas contagens de pulso na entrada do bloco para ser descrito em uma outra aplicação.

7) *Setpoint Ramp Generator (SPG)* – Este bloco é normalmente usado para gerar o ponto de ajuste (*setpoint*) de um bloco PID em aplicações como o controle de temperatura e grupos de reatores.

8) *Input Selector (IS)* – Este bloco fornece uma seleção de até quatro entradas e gera uma saída baseada na ação configurada. Este bloco normalmente deve receber a entrada de um bloco AI ou de outro bloco, e não diretamente do *transducer*.

9) *Arithmetic (AR)* – Este bloco foi projetado para permitir o simples uso de funções matemáticas. O usuário não tem que saber como escrever as equações. O algoritmo de matemática é selecionado por nome, e escolhido pelo usuário para a função a ser realizada.

10) *Timer (TMR)* – Este bloco fornece funções de contagem de tempo incluindo: combinação de múltiplas entradas como OR e AND, medição da duração do sinal discreto de entrada, etc.

11) *Analog Alarm (AAL)* - Este bloco fornece condição de alarme que relata sobre uma saída analógica de qualquer bloco.

E por último, foram definidos mais alguns blocos funcionais para realizarem tarefas com o uso de Controláveis Lógicos Programáveis (CLPs). Alguns destes blocos são:

1) *Multiple Analog Output (MAO)*– Este bloco tem disponível oito subsistemas para entradas de parâmetros. Este bloco funcional tem a mesma característica do bloco Analog output (AO).

2) *Multiple Discret Output (MDO)* - Este bloco tem disponível oito subsistemas para entradas discretas de parâmetros. Este bloco funcional tem a mesma característica do bloco *Discrete Output (DO)*.

3) *Multiple Analog Input (MAI)*- Este bloco tem disponível oito subsistemas para saídas de parâmetros. Este bloco funcional tem a mesma característica do bloco *Analog output (AI)*.

4) *Multiple Discret Input* (MDI) - Este bloco tem disponível oito subsistemas para saídas discretas de parâmetros. Este bloco funcional tem a mesma característica do bloco *Analog DiscretOutput* (DO).

Com o conjunto de vinte e um blocos funcionais adicionados aos blocos definidos para CLPs, prevê que a grande maioria das malhas de controle existentes hoje em PLCs, quando implementadas em sistema de controle *Foundation*, possam ser executadas exclusivamente nos transmissores, e conseqüentemente, de modo realmente distribuído.

6.2 LIMITAÇÕES DOS BLOCOS FUNCIONAIS NA APLICAÇÃO DAS TÉCNICAS DE CONTROLE

Atualmente as estratégias de controle em redes *Fieldbus* realizadas por blocos funcionais executam medições da variável de interesse, decidem se seu valor é aceitável e aplica um esforço corretivo se necessário.

Estas estratégias fazem, geralmente, suas decisões de acordo com o algoritmo (PID) proporcional, integral, derivativa que confia em um único sensor para a medida e um único atuador para a ação corretiva. Esta rotina de controle de única variável trabalha muito bem para uma ampla variedade de problemas de controle comutando as variáveis de processo, que podem ser manipuladas independentemente.

Quando o sistema de controle é requerido para conseguir de uma só vez os objetivos que usam os atuadores múltiplos que afetam todas as variáveis do processo simultaneamente, o problema torna-se muito complicado devido ao fato de não existir nenhuma ferramenta, ou melhor, nenhum bloco funcional, que trate deste problema.

No entanto, aproveitando a flexibilidade da norma *Foundation*, esta tese tem como objetivo desenvolver novos blocos funcionais para solucionar os problemas de controle em *Fieldbus*, que necessitam controlar mais de uma malha de controle para obter uma operação satisfatória do processo.

6.3 PROJETO DOS NOVOS BLOCOS FUNCIONAIS

6.3.1 INTEGRAÇÃO

A implementação de novos Blocos Funcionais deve seguir a metodologia sugerida pela norma *Foundation*. A seguir serão apresentados os passos necessários para o desenvolvimento e implementação dos novos blocos funcionais de acordo com a norma FIELDBUS FOUNDATION - FF890 (1998).

Passo 1:

Inicialmente deve-se modelar o funcionamento do bloco, e a partir deste modelo, identificar as variáveis que serão acessadas por procedimentos de fábrica e aquelas que serão acessadas pelo usuário com o objetivo de diagnóstico, configuração e operação. Pode-se então montar uma lista de variáveis com estas características.

Passo 2:

A lista de variáveis deve ser dividida em três partes de acordo com o seu uso:

- 1 - Variáveis que descrevem o recurso, como espaço de memória e mínimo tempo de execução. Estas variáveis não são editadas pelo usuário;
- 2 - Variáveis configuradas ou verificadas apenas uma vez em fábricas ou no lugar de instalação. Exemplo: unidade de engenharia, *offset* e *span*.
- 3 - Variáveis verificadas periodicamente pelo usuário por procedimentos de manutenção ou operação. Exemplo: alarmes, *setpoints*, sintonias e modo de operação. Neste caso, as variáveis são definidas nos casos 1 e 2, e os parâmetros são definidos como da classe 3.

Passo 3:

As variáveis descrevendo os recursos são parâmetros para o *resource block* do transmissor, além daqueles parâmetros do *resource block* definidos pela norma. As variáveis definidas como sendo de configuração de fábrica ou de uso na instalação do sistema são parâmetros a serem adicionados ao *transducer block*.

Passo 4:

Os parâmetros identificados como sendo de uso periódico, em operação ou em manutenção são parâmetros do futuro bloco funcional.

A norma recomenda que os parâmetros dos novos blocos sejam comparados como os parâmetros dos blocos padrões, em nível de funcionalidade. Caso seja identificada alguma semelhança nos parâmetros do bloco novo com os parâmetros de um bloco padrão, então este deverá servir como base para o projeto, assim bastará adicionar os parâmetros inéditos no bloco padrão. Estes parâmetros são variáveis que devem ser mapeadas como variáveis simples, vetores ou *records*, de acordo como os tipos de dados padrões e então listados em seguida aos parâmetros padrões na definição do bloco.

Os parâmetros de um bloco não padrão são classificados de acordo com a regra acima em parâmetros de *device* normalizados e parâmetros específicos do fabricante.

Caso nenhum bloco padrão sirva de modelo para o novo bloco, os parâmetros deste, caso seja possível, deverão ter as mesmas características dos parâmetros padrões, como nome, função, status e tipo de operação.

A *Fieldbus Foundation* prevê a possibilidade de normalização tanto de parâmetros como de blocos específicos de fabricante caso estes tenham utilidade abrangente.

Na implementação computacional do bloco deve-se criar estruturas que disponibilizem para a camada de aplicação os serviços de inicialização, execução, leitura e escrita para cada parâmetro disponível. Após esta implementação, faz-se a integração destas estruturas, ou métodos, nas estruturas complementares ao *Function block application process* (FBAP): *Device Description* (DD) e ao *Object Dictionary* (OD).

A DD do transmissor deve incluir a descrição do bloco e dos seus parâmetros. Na caracterização dos parâmetros temos: tipo de dado, permissão de acesso, *label*, texto de *help* e a classe dos parâmetros (estático, de entrada ou de saída). Já a descrição do bloco na DD inclui o fabricante, a revisão, todos o parâmetros, as *views* e os grupos de unidades relacionadas.

Quando compilada, a DD fornece dois números inteiros denominados *Member ID* e *Item ID* para cada parâmetro, onde os quais são utilizados na inclusão dos parâmetros do bloco no OD, juntamente com outras informações que caracterizam o parâmetro, entre as quais são mostradas adiante nas Tabelas de Parametrização dos blocos.

Informações sobre o bloco devem estar presente também no arquivo *Capability*, utilizado pelo configurador juntamente com o arquivo de DD para uma correta identificação e configuração de cada diferente transmissor.

No arquivo de *Capability* do transmissor estão as informações de tempo de execução do bloco e quantidade de memória RAM necessárias para armazenar todos os seus parâmetros, assim como o *Item ID* do bloco e o número de parâmetros deste, BRANDÃO (2000).

6.3.2 BLOCO FUNCIONAL OBSERVADOR DE ESTADOS (OE)

6.3.2.1 INTRODUÇÃO

O bloco Observador de Estados (OE) foi desenvolvido de forma que todos os seus parâmetros definidos possam ser suportados pelos dispositivos interoperáveis registrados na *Fieldbus Foundation*.

O projeto deste bloco OE é de ordem plena e foi baseado na teoria de controle moderna apresentada no capítulo 4, com a finalidade de reconstituir os estados de uma planta através das medidas das variáveis de saída e de controle. Este bloco possui também a opção de utilizar o vetor de estado observado na realimentação de estado para gerar um vetor de controle desejado.

O bloco OE foi estruturado para representar tanto modelos de sistemas de uma única entrada e uma única saída (SISO) como modelos de sistemas de múltiplas entradas e múltiplas saídas (MIMO), na qual pode conter um vetor de estado com até seis variáveis de estado para descrever o comportamento dinâmico de um sistema.

6.3.2.2 ESTRUTURA E FUNÇÕES DO BLOCO OE

O algoritmo deste bloco possui um sistema de equações dinâmicas matriciais que são dadas pelas eq.(6.1 a 6.3) e executadas em *loop* num processo de convergência.

$$\dot{\hat{x}} = A_p \hat{x}_p + B_p u + L (y - \hat{y}) \quad (6.1)$$

$$\hat{y} = C_p \hat{x}_p \quad (6.2)$$

$$z = -K \hat{x}_p \quad (6.3)$$

Onde:

(i) A_p é uma matriz estável de ordem $n \times n$, com elementos constantes, chamada matriz de estado. Esta é composta pelos parâmetros da planta real.

(ii) B_p é uma matriz de ordem $n \times q$, com elementos constantes, chamada matriz de entrada. Esta também é composta pelos parâmetros da planta real.

(iii) C_p é uma matriz de ordem $n \times n$, também com elementos constantes chamada matriz de saída.

(iv) u é o vetor do sinal de controle de ordem $q \times 1$, na qual corresponde as entradas de controle do bloco (INc_1, INc_2 e INc_3).

(v) y é o vetor de variáveis medidas da planta de ordem $q \times 1$ na qual corresponde as entradas do bloco (INe_1, INe_2 e INe_3).

(vi) $\hat{y}$ é o vetor de saída dos estados observados da planta de ordem $q \times 1$ no qual será comparado com o vetor y de valores medidos da planta.

(vii) $\hat{x}_p$ é o vetor de estados observados da planta de ordem $n \times 1$.

(viii) z é o vetor de ordem $n \times 1$ na qual corresponde as saídas do bloco (OUTe_1, OUTe_2, OUTe_3, OUTe_4, OUTe_5, OUTe_6).

(ix) L é vetor de ordem apropriada na qual auxiliará a reduzir os efeitos devidos a diferenças entre o modelo dinâmico e a planta real.

(x) K é também é uma matriz de ordem apropriada denominada matriz de ganho de realimentação.

Logo, $n=6$ para termos seis estados disponíveis, assim como citado anteriormente, e $q=3$ de forma que as matrizes fiquem com suas dimensões apropriadas.

Uma representação esquemática do bloco OE, com seu algoritmo, entradas e saídas é apresentada a seguir pela Fig. 6.2:

FIGURA 6.2- Bloco funcional observador de estados

Ao usuário cabe configurar, através dos parâmetros de configuração disponíveis, as matrizes da planta (A_p , B_p e C_p), a matriz do observador (L), as unidades das grandezas de entrada e os ganhos (K) no caso do vetor de estado observado ser utilizado na realimentação de estado para gerar o vetor de controle desejado.

Existem ainda parâmetros de visualização de operação do bloco como: BLOCK_ERR, CONVERGENCE e OUTPUT_ERR.

O conjunto completo de parâmetros do bloco está listado na tabela descritiva a seguir com todas as informações necessárias para a completa caracterização de cada parâmetro:

TABELA 6.1 - Parametrização do bloco Observador de Estados

Índice	Parâmetros	Tipo de Dado (tamanho)	Faixa Válida / Opções	Valor Defaut	Unid.	M.A	Descrição
1	St_rev	Unsigned (16)		0	Na	S	Revisão estática do bloco

2	Tag_desc	OctString (32)		Espaços	Na	S	Rótulo descritivo
3	Strategy	Unsigned (16)		0	Na	S	Identificador de grupos de blocos
4	Alert_key	Unsigned (8)	1 a 255	0	Na	S	Identificador da unidade da planta
5	Mode_blk	DS-69	Auto/OOS Man	OOS	E	SS	Relação dos modos permitidos e atual
6	Block_err	BitString (2)			E	D/ RO	Status de erro do bloco
7	INm_1	DS-65			IN	D	Entrada da variável medida da planta
8	INm_2	DS-65			IN	D	Entrada da variável medida da planta
9	INm_3	DS-65			IN	D	Entrada da variável medida da planta
10	INc_1	DS-65			IN	D	Entrada da variável controle
11	INc_2	DS-65			IN	D	Entrada da variável controle
12	INc_3	DS-65			IN	D	Entrada da variável controle
13	Output_err	Float			Na	D/RO	Diferença entre a saída da planta real e a saída do modelo
14	$A_{p11} \dots A_{p16}$ $\vdots$ $A_{p61} \dots A_{p66}$	36 Float		0	Na	S	Elementos da matriz A_p da planta
15	$B_{p11} \dots B_{p13}$ $\vdots$ $B_{p32} \dots B_{p33}$	18 Float		0	Na	S	Elementos da matriz B_p da planta
16	$C_{p11} \dots C_{p16}$ $\vdots$ $C_{p61} \dots C_{p66}$	36 Float		0	Na	S	Elementos da matriz C_p da planta
17	$L_{11} \dots L_{17} \dots L_{11}$	3 Float		1	Na	S	Elementos da matriz L do observador
18	$K_{11} \dots K_{16}$	6 Float		1			Elementos da matriz K de realimentação
19	Pré X_{p1}	Float		0	Na	D/RO	Valor do estado 1
20	Pré X_{p2}	Float		0	Na	D/RO	Valor do estado 2
21	Pré X_{p3}	Float		0	Na	D/RO	Valor do estado 3
22	Pré X_{p4}	Float		0	Na	D/RO	Valor do estado 4
23	Pré X_{p5}	Float		0	Na	D/RO	Valor do estado 5
24	Pré X_{p6}	Float		0	Na	D/RO	Valor do estado 6
25	OUTe_1	DS-65			Na	D/ Man	Valor do estado 1 multip. Por K_1
26	OUTe_2	DS-65			Na	D/ Man	Valor do estado 2 multip. Por K_2
27	OUTe_3	DS-65			Na	D/ Man	Valor do estado 3 multip. Por K_3
28	OUTe_4	DS-65			Na	D/ Man	Valor do estado 4 multip. Por K_4

29	OUTe_5	DS-65			Na	D/ Man	Valor do estado 5 multip. Por K_5
30	OUTe_6	DS-65			Na	D/ Man	Valor do estado multip. Por K_6
31	Convergence	Unsigned (8)	1=not conv. 2=conv.	1	E	S	O identificador de convergência do observador
32	INm_1,2,3 UNITS	Unsigned (16)		0	IN	S	Unidade de eng. da entrada IN para o bloco display
33	INc_1,2,3 UNITS	Unsigned (16)		0	IN	S	Unidade de eng. da entrada IN para o bloco display
34	Update_Evt	DS-73			Na	D	Alerta gerado pela alteração de parâmetros
35	Grant_Deny	DS-70		0	Na	D	Controle de acesso para aplicações remotas

Legenda: M.A - Modo de armazenamento

E - Parâmetro *enumerated*

Na - Parâmetro adimensional

RO – Somente leitura

D – Dinâmico

N – Não - volátil

S – Estático

O bloco possui seis entradas (INp_1, INp_2, INp_3, INc_1, INc_2 e INc_3) e seis saídas (OUTr_1, OUTr_2, OUTr_3, OUTr_4, OUTr_5, OUTr_6).

As três primeiras entradas são valores medidos da planta real e as outras três são sinais de controle recebidos de outros blocos.

As saídas deste bloco são os valores dos estados observados da planta multiplicados pelos ganhos de realimentação ($K_{1...6}$). O valor *default* deste parâmetro $K_{1...6}$ é 1. No caso da estratégia de controle utilizar o bloco OE somente para observar os estados, com o valor do *default* de 1, as saídas do bloco corresponderam conseqüentemente os estados observados da planta.

Com relação aos modos de operação, o bloco suporta três. Sendo esses:

(1) Modo OOS (*Out of Service*): o bloco fica “desligado”;

(2) Modo Auto (Automático): a saída desejada é obtida através do algoritmo definido pelo bloco;

(3) Modo Man (Manual): as saídas desejadas são diretamente definidas pelo usuário, sendo que o *default* destas saídas em modo manual é zero.

Em operação Man ou Auto, o bloco deve recuperar os valores de Pre_Xp_1 , Pre_Xp_2 , Pre_Xp_3 , Pre_Xp_4 , Pre_Xp_5 e Pre_Xp_6 da execução anterior para serem utilizados na execução seguinte.

O parâmetro CONVERGENCE indica se o sistema de equações do bloco esta respondendo da mesma forma que a planta, ou seja, se a saída do modelo convergiu ou não.

O bloco também possui uma chave (*State Selector*) para selecionar os estados apropriados para serem comparados com os valores medidos recebidos de outros blocos.

Com relação aos *status* das entradas, estes não afetam a execução do algoritmo, sendo que os *status* de Xp_1 , Xp_2 , Xp_3 , Xp_4 , Xp_5 e Xp_6 são semelhante ao pior status entre as entradas. Este tipo de implementação garante o mecanismo da norma *Foundation* de propagação de *status* na malha, como ferramenta para garantia de segurança do sistema em caso de falhas ou emergência.

As condições de execução do algoritmo não mudam o modo do bloco, nem com as entradas em status *Bad*. Caso ocorra uma operação indefinida, a operação do bloco não é abortada.

No caso de perda de energia no transmissor, todos os parâmetros de configuração são armazenados em memória não volátil, garantindo assim uma correta reinicialização da operação normal do bloco assim que o transmissor é novamente ligado, sem a necessidade de reconfiguração.

O sistema de alarme do Bloco OE segue os requisitos básicos da *Foundation*, portanto o bloco pode ser perfeitamente integrado a qualquer tipo de malha de controle.

6.3.3 BLOCO FUNCIONAL ADAPTATIVO POR MODELO DE REFERÊNCIA (AMR)

6.3.3.1 INTRODUÇÃO

O bloco funcional Adaptativo por Modelo de Referência (AMR) foi projetado dentro dos padrões da norma *Fieldbus Foundation*, baseado no algoritmo de controle adaptativo direto, na qual utiliza variáveis de estado para descrever o conjunto de equações do sistema adaptativo por modelo de referência.

O bloco AMR tem como objetivo encontrar a lei de controle adaptativa de realimentação que muda a estrutura e a dinâmica da planta de forma que suas propriedades são exatamente as mesmas do que as de um modelo de referência, ou seja, o desempenho desejado da planta é expresso em termos de um modelo de referência que fornece a resposta desejada para um determinado sinal de comando.

6.3.3.2 ESTRUTURA E FUNÇÕES DO BLOCO AMR

O algoritmo deste bloco é baseado no esquema adaptativo por modelo de referência com ajuste de parâmetros pela teoria de *Lyapunov*, assim como descrito no capítulo 5. O bloco possui um conjunto de equações matriciais que são mostradas novamente nas eq.(6.4 a 6.9) e executadas em *loop* para obter a lei de controle adaptativa. A eq.(6.6) representa a lei de controle adaptativa, as eq.(6.7) e (6.8) são termos que compõem esta lei e a eq.(6.9) é a equação do erro, na qual é definido pela diferença entre os estados observados da planta real e os estados do modelo referência.

$$\dot{\mathbf{x}}_m = \mathbf{A}_m \mathbf{x}_m + \mathbf{B}_m \mathbf{r} \quad (6.4)$$

$$\mathbf{y}_m = \mathbf{C}_m \mathbf{x}_m \quad (6.5)$$

$$\mathbf{u} = \mathbf{L} \mathbf{K} \hat{\mathbf{x}}_p + \mathbf{L} \mathbf{r} \quad (6.6)$$

$$\dot{\mathbf{K}} = -\mathbf{B}_m^T \mathbf{P} \mathbf{e} \hat{\mathbf{x}}_p^T \quad (6.7)$$

$$\dot{\mathbf{L}} = -\mathbf{L} \mathbf{B}_m^T \mathbf{P} \mathbf{e} \mathbf{u}^T \mathbf{L} \quad (6.8)$$

$$\mathbf{e} = \hat{\mathbf{x}}_p - \mathbf{x}_m \quad (6.9)$$

Onde:

(i) A_m é uma matriz estável de ordem $n \times n$, com elementos constantes, chamada matriz de estado. Esta é composta pelos os parâmetros do modelo de referência do sistema.

(ii) B_m é uma matriz de ordem $n \times q$, com elementos constantes chamada matriz de entrada. Esta também é composta pelos parâmetros do modelo de referência.

(iii) C_m é uma matriz de ordem $n \times n$, também com elementos constantes chamada matriz de saída.

(iv) r é vetor de referência de entrada de ordem $q \times 1$, na qual corresponde as entradas do bloco (INr_1, INr_2 e INr_3).

(v) x_m é o vetor de estados de ordem $n \times 1$, que fornece a trajetória desejada na qual a planta deve seguir.

(vi) y_m é o vetor de saída de ordem $n \times 1$ do modelo referência na qual corresponde as saídas do modelo de referência.

(vii) u é a lei de controle adaptativa na qual consiste de um vetor de ordem $q \times 1$, que corresponde as saídas do bloco (OUT_1, OUT_2 e OUT_3).

(viii) u^T é o vetor da lei de controle adaptativa transposto, de ordem $1 \times q$.

(iix) P é uma matriz de ordem $n \times n$, obtida através da seguinte equação *Lyapunov*: $PA_m + A_m^T P = -Q$.

(ix) B_m^T é matriz de entrada do modelo de referência transposta. Esta matriz é composta por elementos constantes de ordem $q \times n$.

(x) e é um vetor de ordem $q \times 1$, chamado vetor de erro.

(xi) $\hat{x}_p$ é o vetor de estados observados da planta de ordem $n \times 1$.

(xii) $\hat{x}_p^T$ é o vetor de estados observados da planta transposto de ordem $1 \times n$.

Sendo que $n = 6$ e $q = 3$, assim conforme definido anteriormente.

Uma representação esquemática do bloco AMR, com seu algoritmo, entradas e saídas é apresentada a seguir pela Fig.(6.3):

FIGURA 6.3 - Bloco funcional adaptativo por modelo de referência

Ao usuário cabe configurar, através dos parâmetros de configuração disponíveis, as matrizes do modelo de referência (A_m , B_m e C_m) e as unidades das grandezas de entrada. Também devem ser configuradas as matrizes (P , B_m^T), as condições iniciais das integrais e as unidades das grandezas de entrada.

O conjunto completo de parâmetros do bloco está listados na tabela descritiva a seguir com todas as informações necessárias para a completa caracterização de cada parâmetro:

TABELA 6.2 - Parametrização do bloco Adaptativo por Modelo de Referência

Índice	Parâmetros	Tipo de Dado (tamanho)	Faixa Válida / Opções	Valor Defaut	Unid.	MA	Descrição
1	St_rev	Unsigned (16)		0	Na	S	Revisão estática do bloco
2	Tag_desc	OctString (32)		Espaços	Na	S	Rótulo descritivo
3	Strategy	Unsigned		0	Na	S	Identificador de

		(16)					grupos de blocos
4	Alert_key	Unsigned (8)	1 a 255	0	Na	S	Identificador da unidade da planta
5	Mode_blk	DS-69	Auto/OOS Man	OOS	E	SS	Relação dos modos permitidos e atual
6	Block_err	BitString (2)			E	D/RO	Status de erro do bloco
7	INr_1	DS-65			IN	D	Entrada da variável 1 de referência
8	INr_2	DS-65			IN	D	Entrada da variável 2 de referência
9	INr_3	DS-65			IN	D	Entrada da variável 3 de referência
10	INp_1	DS-65			IN	D	Entrada do estado 1 observ. da planta
11	INp_2	DS-65			IN	D	Entrada do estado 2 observ. da planta
12	INp_3	DS-65			IN	D	Entrada do estado 3 observ. da planta
13	INp_4	DS-65			IN	D	Entrada do estado 4 observ. da planta
14	INp_5	DS-65			IN	D	Entrada do estado 5 observ. da planta
15	INp_6	DS-65			IN	D	Entrada do estado 6 observ. da planta
16	$A_{m11} \dots A_{m16}$ $\vdots$ $A_{m61} \dots A_{m66}$	36 Float		0	Na	S	Elementos da matriz A_m do modelo
17	$B_{m11} \dots B_{m13}$ $\vdots$ $B_{m32} \dots B_{m33}$	18 Float		0	Na		Elementos da matriz B_m do modelo
18	$C_{m11} \dots C_{m16}$ $\vdots$ $C_{m61} \dots C_{m66}$	36 Float		0	Na	S	Elementos da matriz C_m do modelo
19	$B_{m11}^T \dots B_{m16}^T$ $\vdots$ $B_{m31}^T \dots B_{m36}^T$	18 Float		0	Na	S	Elementos da matriz B_m^T do modelo
20	$P_{11} \dots P_{16}$ $\vdots$ $P_{61} \dots P_{66}$	36 Float		0	Na	S	Elementos da matriz P obtida por <i>Lyapunov</i>
21	CI L	18 Float			Na	D	Condições iniciais da integral L
22	CI K	36 Float			Na	D	Condições iniciais da integral L
23	OUTr_1	DS-65			Na	D/Man	Valor referência do estado 1
24	OUTr_2	DS-65			Na	D/Man	Valor referência do estado 2
25	OUTr_3	DS-65			Na	D/Man	Valor referência do estado 3
26	OUTr_4	DS-65			Na	D/Man	Valor referência do estado 4

27	OUTr_5	DS-65			Na	D/ Man	Valor referência do estado 5
28	OUTr_6	DS-65			Na	D/ Man	Valor referência do estado 6
29	OUT_1	DS-65			Na	D/ Man	Valor do estado 1 multiplicado pelo ganho K_1
30	OUT_2	DS-65			Na	D/ Man	Valor do estado 2 multiplicado pelo ganho K_2
31	OUT_3	DS-65			Na	D/ Man	Valor do estado 3 multiplicado pelo ganho K_3
32	INr_1,2,3 UNITS	Unsigned (16)		0	IN	S	Unidade de eng. da entrada IN para o bloco display
33	INm_1,2,3, 4,5,6 UNITS	Unsigned (16)		0	IN	S	Unidade de eng. da entrada IN para o bloco display
34	INp_1,2,3, 4,5,6 UNITS	Unsigned (16)		0	IN	S	Unidade de eng. da entrada IN para o bloco display
35	Update_Evt	DS-73			Na	D	Alerta gerado pela alteração de parâmetros
36	Grant_Deny	DS-70		0	Na	D	Controle de acesso para aplicações remotas

Legenda: M.A - Modo de armazenamento

E - Parâmetro *enumerated*

Na - Parâmetro adimensional

RO – Somente leitura

D – Dinâmico

N – Não - volátil

S – Estático

O bloco possui nove entradas (INr_1, INr_2, INr_3, INp_1, INp_2, INp_3, INp_4, INp_5, INp_6) e nove saídas (OUT_1, OUT_2, OUT_3, OUTr_1, OUTr_2, OUTr_3, OUTr_4, OUTr_5, OUTr_6).

As entradas correspondem aos pontos de operação do sistema (*setpoint*), e aos estados observados da planta, respectivamente. As três primeiras saídas compõem a lei de controle adaptativa e as outras seis correspondem aos estados do modelo de referência.

Os modos de operação e as condições de execução deste bloco são iguais ao do bloco OE, assim como citado anteriormente.

O sistema de alarme do Bloco AMR também segue os requisitos básicos da *Foundation*, portanto o bloco pode ser perfeitamente integrado a qualquer tipo de malha de controle.

6.4 MALHAS DE CONTROLE NO *FIELDBUS*

As malhas de controle no *Fieldbus* consistem no interligamento lógico de diversos blocos funcionais implementados nos dispositivos (*devices*) da rede através de um *software* configurador. A conexão entre os blocos é definida como um *link* da rede. A Fig.(6.4) mostra um simples exemplo de malha de controle com os respectivos *links* interligando os blocos funcionais.

FIGURA 6.4 - Exemplo de uma malha de controle

6.4.1 MALHA DE CONTROLE COM REALIMENTAÇÃO DE ESTADO

As malhas de controle por realimentação de estado utilizam o novo bloco Observador de Estados (OE) e alguns blocos padrões como: Entrada Analógica (AI), Saída Analógica (AO), Aritmético (ARTH), e Controlador Proporcional Integral Derivativo (PID). Além destes blocos as malhas também utilizam o bloco Constante (CT) que é um bloco não padrão definido pela SMAR, na qual possui seis saídas analógicas e duas digitais constantes e configuráveis pelo usuário.

A configuração dos blocos de cada uma das malhas é variável, pois para cada tipo de grandeza física como: parâmetros de escalas, limites e alarmes são diferentes. No entanto, alguns parâmetros de configuração dos blocos não mudam quaisquer que sejam as aplicações. Nestes parâmetros se definem as operações da malha e sua configuração básica.

6.4.1.1 MALHA DE CONTROLE POR OBSERVAÇÃO DE ESTADOS

A malha de controle por realimentação de estado consiste: O sinal de referência (*Setpoint*) é gerado pelo bloco funcional “Constante”(CT). O bloco Observador de Estados (OE) recebe as variáveis de entrada e a de saída do sistema real através do bloco de Entrada Analógica (AI) e do bloco Saída Analógica (AO), respectivamente. Esse bloco estima os estados da planta real, multiplica pelos ganhos de realimentação e fornece os resultados ao bloco Aritmético (ARTH).

O bloco Aritmético (ARTH) tem a função de somar os estados multiplicados pelos ganhos de realimentação e comparar com o valor de referência fornecido pelo bloco constante (CT), determinando o erro. Este bloco envia o erro ao bloco saída analógica (AO) para fazer as correções necessárias e atuar na planta real. A malha de controle por estimação de estado, conforme apresentada no capítulo 4 pela Fig. 4.1, é mostrada a seguir na Fig. 6.5.

FIGURA 6.5 - Malha de controle por observação de estados

6.4.1.2 MALHA DE CONTROLE COM OBSERVADOR AUTÔNOMO

A malha de controle com observador autônomo segue os mesmos princípios de funcionamento da malha de controle por realimentação de estados, no entanto a entrada (IN_1) do bloco Observador é obtida diretamente da saída do bloco Aritmético (ARTH). A malha de controle com observador autônomo ou seguidor de erro, conforme apresentada no capítulo 4 pela Fig. 4.2, é mostrada a seguir na Fig.6.6.

FIGURA 6.6 - Malha de controle com observador autônomo

6.4.1.3 MALHA DE CONTROLE COM OBSERVADOR AUTÔNOMO E REALIMENTAÇÃO DE SAÍDA

A malha de controle com observador autônomo e realimentação de saída, conforme apresentada no capítulo 4 pela Fig. 4.3, é mostrada a seguir na Fig. 6.7.

FIGURA 6.7 - Malha de controle com observador autônomo e realimentação de saída

Esta malha de controle utiliza o bloco PID para gerar um valor de referência na qual será comparado com os estados observados e multiplicados pelos ganhos de realimentação. Através desta comparação é obtido o erro do sistema, que é enviado ao bloco OE para realimentação e para o bloco AO para fazer as correções necessárias na planta real.

6.4.1.4 MALHA DE CONTROLE COM ALOCAÇÃO DE ZEROS NO OBSERVADOR ESTADOS

A malha de controle com alocação de zero e observador, conforme apresentada no capítulo 4 pela Fig. 4.12, é mostrada a seguir na Fig. 6.8.

FIGURA 6.8 - Malha de controle com alocação de pólos e observador

6.4.1.5 MALHA DE CONTROLE ADAPTATIVA POR MODELO REFERÊNCIA

A malha de controle adaptativa por modelo de referência consiste da seguinte forma: O bloco observador de estados (OE), que recebe os valores das variáveis de entrada e saída do sistema real através dos blocos entrada analógica (AI) e saída analógica (AO), observa os estados e fornece a saída observada da planta ao bloco de controle adaptativo por modelo de referência (AMR). O bloco AMR que recebe a saída estimada da planta do bloco observador de estados e o *setpoint* do bloco constante (CT), compara as saídas (desejada e observada) determinando o erro e, através de um mecanismo de adaptação, produz a correção e a envia ao bloco ao para atuar na planta real. Na Fig.(6.9) é mostrada a malha de controle proposta.

FIGURA 6.9 - Malha de controle adaptativa

Capítulo 7

Testes e Resultados

Neste capítulo apresentam os resultados dos testes obtidos pelos blocos funcionais desenvolvidos.

7.1 INTRODUÇÃO

São apresentados neste capítulo os resultados dos testes obtidos pelos novos funcionais desenvolvidos (bloco funcional observador de estados e bloco funcional controle adaptativo modelo de referência) para várias estratégias de controle, assim como mostradas no capítulo 6.

Parte dos testes foram realizados através de um experimento prático utilizando um sistema térmico, e a outra parte foi simulada no ambiente *Matlab*.

Embora uma série muito extensa de testes tenha sido realizada para verificar o desempenho dos blocos funcionais, somente os resultados principais serão apresentados.

7.2 RESULTADOS PRÁTICOS

7.2.1 DESCRIÇÃO DO SISTEMA FÍSICO

Na realização dos testes de campo para verificar o funcionamento e o desempenho do bloco funcional Observador de Estados (OE) foi utilizado um sistema térmico de controle de temperatura. Este sistema dinâmico utilizado é composto por um forno elétrico instrumentado com transmissores *Fieldbus Foundation*, um transmissor de temperatura (TT 302) e um conversor *Fieldbus* 4 a 20 mA (FI 302), ambos da Smar. Foi também utilizado para auxiliar no controle da planta um medidor Hp, uma fonte de corrente chaveada, uma fonte de tensão e um microcomputador do tipo PC. A Fig.7.1 mostra o esquema da montagem do sistema.

FIGURA 7.1 - Sistema térmico instrumentado com *Fieldbus*

A resistência do forno está ligada ao transmissor de temperatura através de um termopar e os dois transmissores estão em um canal do *Fieldbus* ligado a uma placa de interface PCI302 da Smar que está acoplada ao microcomputador. Esta placa PCI é o LAS (*Link Active Scheduler*) do canal *Fieldbus*.

7.2.2 DESCRIÇÃO DO FUNCIONAMENTO DO SISTEMA TÉRMICO

O transmissor de temperatura está ligado a um termopar tipo K, que mede a temperatura interna do forno. A compensação de junta fria é realizada pelo transmissor através da leitura de um termistor Pt100 à temperatura ambiente. O conversor de *Fieldbus* para 4 a 20 mA está acionando uma chave estática que alimenta a resistência elétrica e fornece potência ao forno. O sistema dinâmico utilizado possui entrada em porcentagem de potência, onde vale a relação proporcional direta entre 0 a 100% e 4 a 20 mA, e saída em temperatura (°C).

7.2.3 CONFIGURAÇÃO E SUPERVISÃO DO SISTEMA TÉRMICO

Para configuração do sistema foi utilizado o *software Syscon*. A Fig.7.2 mostra a janela do *Syscon* com a configuração do canal utilizado.

FIGURA 7.2 - Canal *Fieldbus* no configurador *Syscon*

Através do *Syscon* foi realizada a configuração dos equipamentos de campo e da estratégia de controle onde foram utilizados os blocos funcionais utilizados pelos dispositivos *Fieldbus*. As configurações foram criadas e editadas graficamente de maneira semelhante a um fluxograma. Pelo *Syscon* também foi acessado os dados dos equipamento de campo, incluindo os parâmetros dos blocos funcionais, blocos *transdutores* e blocos de recursos.

O sistema de supervisão foi realizado pelo *software Aimax* que permitiu monitorar e atuar no sistema em funcionamento e disponibilizar ferramentas de criação de telas de sinótico, registros gráficos, tratamento de alarmes, relatórios, bancos de dados, transferência de informações para outros aplicativos, visualização em múltiplos monitores, dentre as muitas outras características.

7.2.4 MODELAGEM DA PLANTA

A função de transferência da planta em questão foi obtida experimentalmente aplicando-se uma entrada em degrau de 60 % no sistema em malha aberta, e obtendo a curva de resposta. Na qual pode ser vista pela Fig. 7.3:

FIGURA 7.3 - Curva real da resposta para uma entrada degrau de 60%

Assumindo que a curva de resposta transitória vista pela Fig. 7.3 é dada por uma soma de exponenciais, descrita pela seguinte equação:

$$y(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t} + C_3 e^{\lambda_3 t} \quad \text{eq.(7.1)}$$

Através de uma interpolação dos dados reais coletados foram levantados os coeficientes da eq.(7.1). Logo os valores dos coeficientes são: $C_1 = 184,04$; $C_2 = -46,06$; $C_3 = -150,47$; $\lambda_1 = 0$; $\lambda_2 = -5,19 \times 10^{-3}$; $\lambda_3 = -1,14 \times 10^{-3}$.

Com os respectivos coeficientes encontrados através da curva de dados reais a eq.(7.1) torna-se:

$$y(t) = 184,04 - 46,06 e^{-5,19 \times 10^{-3} t} - 150,06 e^{-1,14 \times 10^{-3} t} \quad \text{eq.(7.2)}$$

A Fig. 7.4 mostra a curva de resposta para uma entrada degrau de 60%, obtida proveniente da eq.(7.2) obtida através da soma de exponenciais.

FIGURA 7.4 - Curva de resposta para uma entrada degrau de 60% da curva obtida

Uma comparação entre a curva de resposta obtida e a curva real é mostrada pela Fig. 7.5:

FIGURA 7.5 - Curva dos dados reais e a curva obtida.

Dividindo os coeficientes c_1 , c_2 e c_3 da equação (7.2) por 60, obtemos a equação da curva de resposta normalizada para uma entrada degrau unitária. Logo temos:

$$y_n(t) = 3,067 - 0,768 e^{-5,19 \cdot 10^{-3} t} - 2,507 e^{-1,14 \cdot 10^{-3} t} \quad \text{eq.(7.3)}$$

A curva de resposta normalizada para uma entrada degrau unitária pode ser vista pela Fig. 7.6.

FIGURA 7.6 - Curva de resposta normalizada para uma entrada degrau unitário

Aplicando-se a Transformada de Laplace na eq.(7.3) temos:

$$Y(s) = \frac{3,067}{s} - \frac{0,768}{s + 5,19 \times 10^{-3}} - \frac{2,507}{s + 1,14 \times 10^{-3}} \quad \text{eq.(7.4)}$$

Admitindo-se que a Transformada de Laplace da resposta tem a seguinte forma:

$$Y(s) = \frac{a_1 s + a_0}{s(s + p_1)(s + p_2)} \quad \text{eq.(7.5)}$$

Expandindo a eq.(7.5) em frações parciais temos:

$$Y(s) = \frac{r_1}{s} + \frac{r_2}{(s + p_1)} + \frac{r_3}{(s + p_2)} \quad \text{eq.(7.6)}$$

Onde:

$$\begin{aligned} r_1 &= \left[\frac{a_1 s + a_0}{s(s + p_1)(s + p_2)} \times s \right]_{s=0} = \frac{a_0}{p_1 p_2} \\ r_2 &= \left[\frac{a_1 s + a_0}{s(s + p_1)(s + p_2)} \times (s + p_1) \right]_{s=-p_1} = \frac{a_0 - p_1 a_1}{p_1(p_1 - p_2)} \\ r_3 &= \left[\frac{a_1 s + a_0}{s(s + p_1)(s + p_2)} \times (s + p_2) \right]_{s=-p_2} = \frac{a_0 - p_2 a_1}{p_2(p_2 - p_1)} \end{aligned} \quad \text{eq.(7.7)}$$

Comparando a eq.(7.6) com a eq.(7.4) temos que: $r_1 = 3,067$; $r_2 = -0,768$;

$r_3 = -2,507$; $p_1 = 5,19 \times 10^{-3}$ e $p_2 = 1,14 \times 10^{-3}$.

Substituindo estes valores na eq.(7.7) obtemos: $a_0 = 1,8091 \times 10^{-5}$ e $a_1 = 1,3236 \times 10^{-2}$.

Portanto a Transformada de Laplace da resposta torna-se:

$$Y(s) = \frac{1,32 \times 10^{-2} s + 1,81 \times 10^{-5}}{s(s + 5,19 \times 10^{-3})(s + 1,14 \times 10^{-3})} \quad \text{eq.(7.8)}$$

No entanto, temos que a Função de Transferência de alimentação direta é representada pela figura a seguir:

FIGURA 7.7 - Função de Transferência de alimentação direta.

A partir da Fig. 7.7 temos que:

$$G(s) = \frac{Y(s)}{U(s)} \quad \text{eq.(7.9)}$$

Como temos a entrada degrau unitário, onde:

$$U(s) = \frac{1}{s} \quad \text{eq.(7.10)}$$

Substituindo a eq.(7.7) e a eq.(7.9) na eq.(7.8) obtemos a Função de Transferência do sistema térmico:

$$G(s) = \frac{1,32 \times 10^{-2} s + 1,81 \times 10^{-5}}{s (s + 5,19 \times 10^{-3})(s + 1,14 \times 10^{-3})} \quad \text{eq.(7.11)}$$

Portanto a Função de Transferência do sistema térmico é :

$$G(s) = \frac{1,32 \times 10^{-2} s + 1,81 \times 10^{-5}}{(s + 5,19 \times 10^{-3})(s + 1,14 \times 10^{-3})} \quad \text{eq.(7.12)}$$

7.2.4.1 INCLUSÃO DO TEMPO MORTO NA FUNÇÃO DE TRANSFERÊNCIA

O tempo morto ou atraso de transporte existe na maioria dos sistemas de controle de processo. Este representa um atraso na medida, isto é, um atraso na ação do controlador ou um atraso na operação do atuador. A função de transferência do tempo morto adotado neste caso é dada por:

$$\frac{Y(s)}{X(s)} = e^{-T_s s} \quad \text{eq.(7.13)}$$

Se o Tempo Morto for muito pequeno em relação às constantes de tempo presentes na resposta, então $e^{-T s}$ pode ser aproximado por:

$$e^{-T s} \cong \frac{1}{T s + 1} \quad \text{eq.(7.14)}$$

Portanto, para o sistema térmico em questão encontramos $T=15,579$ seg. Logo temos:

$$e^{-15,579 s} \cong \frac{1}{15,79 s + 1} \quad \text{eq.(7.15)}$$

Incluindo o tempo morto em $G(s)$, obtemos a função de transferência final do sistema térmico:

$$G(s) = \frac{0,85 \times 10^{-3} (s + 1,37 \times 10^{-3})}{(s + 64,19 \times 10^{-3})(s + 5,19 \times 10^{-3})(s + 1,14 \times 10^{-3})} \quad \text{eq.(7.16)}$$

7.2.5 MODELAGEM POR ESPAÇO DE ESTADOS DO SISTEMA TÉRMICO

Assim como visto anteriormente no capítulo 4, item 4.3.1, a representação de um sistema dinâmico na forma de espaço de estado é realizada através das equações (7.17), equação de estado, e (7.18), equação de saída:

$$\dot{x} = Ax + Bu \quad \text{eq.(7.17)}$$

$$y = Cx \quad \text{eq.(7.18)}$$

Transformando a equação (7.16) na forma de espaço de estado obtemos as matrizes da equação de estado e da equação de saída:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3,786 \times 10^{-7} & -4,122 \times 10^{-4} & -7,052 \times 10^{-2} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 8,496 \times 10^{-4} \\ -5,875 \times 10^{-5} \end{bmatrix} \quad \text{e} \quad C = [1 \quad 0 \quad 0].$$

7.2.6 CONFIGURAÇÃO DO MODELO DA PLANTA E DOS PARÂMETROS DE CONTROLE NO BLOCO FUNCIONAL OBSERVADOR DE ESTADOS

Assim como apresentado no capítulo 6, item 6.3.3.2, o bloco funcional Observador de Estados (OE) tem a sua estrutura mostrada pela Fig. 6.2, na qual precisamos determinar os parâmetros das matrizes A_p , B_p , C_p , K e L .

7.2.6.1 DETERMINAÇÃO DOS PARÂMETROS DA PLANTA

Para a implementação do modelo da planta no bloco funcional Observador de Estados é necessário converter o modelo do domínio contínuo no tempo para o discreto, ou seja, é necessário discretizar as matrizes A, B e C. Esta conversão deve ser realizada com a inclusão de um *holder* de ordem zero. O tempo de amostragem do sistema digital deve ser calculado com base no macrociclo da estratégia de controle utilizada no *Fieldbus*, neste caso 550 ms.

O período de 550 ms de macrociclo se adequa perfeitamente ao sistema dinâmico controlado, que apresenta uma velocidade relativamente menor. Desta maneira não se caracteriza a influência da discretização do controle no desempenho deste, ou seja, o sistema de aquisição e controle pode ser considerado contínuo.

Durante o período do macrociclo, o bloco observador de estados é executado uma única vez, porém o modelo dinâmico interno realiza cinco *loops* a cada execução, portanto o valor de tempo do macrociclo deve se dividido por cinco para se achar o correto valor do período de amostragem para a discretização.

Discretizando-se o modelo descrito em A, B e C com um *holder* de ordem zero e taxa de amostragem de 110 ms, temos as seguintes matrizes dinâmicas discretas no tempo:

$$A_p = \begin{bmatrix} 1,000 \times 10^0 & 1,100 \times 10^{-1} & 6,030 \times 10^{-3} \\ -2,286 \times 10^{-9} & 1,000 \times 10^0 & 1,100 \times 10^{-1} \\ -4,150 \times 10^{-8} & -4,520 \times 10^{-5} & 9,920 \times 10^{-1} \end{bmatrix},$$

$$B_p = \begin{bmatrix} 5,130 \times 10^{-6} \\ 9,310 \times 10^{-5} \\ -6,440 \times 10^{-6} \end{bmatrix} \quad \text{e} \quad C_p = [1 \quad 0 \quad 0].$$

7.2.6.2 DETERMINAÇÃO DO PARÂMETRO K DE REALIMENTAÇÃO

Assim, com o visto anteriormente no capítulo 4, item 4.32, a lei de controle de realimentação de estados é dada por:

$$u = -Kx \quad \text{eq.(7.19)}$$

Substituindo o sinal de controle da Eq. (7.19) na Eq. (7.17) tem-se:

$$\dot{x} = (A - B K) x \quad \text{eq.(7.20)}$$

A matriz de ganho de realimentação de estado (K) que força os autovalores $A - BK$ a serem $\mu_1, \mu_2, \dots, \mu_n$ (valores desejados) é determinada para um tempo de acomodação de aproximadamente 350s e um *overhoot* $\geq 5\%$. Logo as localizações dos pólos desejados de malha fechada foram: $\mu_1 = -1,3668 \times 10^{-3}$, $\mu_2 = -1,7000 \times 10^{-2}$ e $\mu_3 = -4,9771 \times 10^{-2}$.

Desta forma, se obteve a matriz K para ser: $K_1 = 9,9585 \times 10^{-1}$, $K_2 = 7,8588 \times 10^{-1}$ e $K_3 = 1,1770 \times 10^{-3}$.

7.2.6.3 DETERMINAÇÃO DO PARÂMETRO L DO OBSERVADOR

Neste mesmo aspecto os pólos desejados do observador de estado foram projetados para serem aproximadamente 8 vezes mais rápido do que os pólos desejados de K . Portanto $\mu_{e1} = -8,000 \times 10^{-2}$, $\mu_{e2} = -9,000 \times 10^{-2}$ e $\mu_{e3} = -1,000 \times 10^{-1}$. Logo a matriz L calculada foi: $L_1 = 1,9948 \times 10^{-1}$, $L_2 = 9,720 \times 10^{-3}$ e $L_3 = -4,48092 \times 10^{-5}$.

7.2.7 ESTRATÉGIA DE CONTROLE NO *FIELD*BUS COM O BLOCO FUNCIONAL OBSERVADOR DE ESTADOS

A malha de controle será implementada conforme a Fig. 7.8 a seguir.

FIGURA 7.8 - Diagrama de blocos da planta e do observador de estado

No *Fieldbus* a malha da estratégia de controle será configurada conforme a ligação dos blocos funcionais mostrada a seguir pela Fig. 7.9:

FIGURA 7.9 - Estratégia de controle com o bloco Observador de Estados

O sinal de referência (*setpoint*) na malha de controle acima é gerado pelo bloco funcional Constante (CT-FI302). O bloco observador de estados (EE-FI302) recebe a variável de entrada (temperatura em °C) e a variável de saída (corrente de 4 a 20mA) do sistema real através dos blocos entrada analógica (AI-TT302) e saída analógica (AO-FI302) respectivamente. Então são observados os estados da planta real, multiplicados pelos ganhos de realimentação e fornecidos os resultados ao bloco aritmético (ARTH-FI302). O bloco aritmético tem a função de somar os estados multiplicados pelos ganhos de realimentação e compará-los com o valor de referência fornecido pelo bloco constante, determinando o erro e enviando ao bloco de saída analógica para fazer as correções necessárias e atuar na planta real.

7.2.8 RESULTADOS OBTIDOS

Na Fig. 7.10, Fig. 7.11 e Fig. 7.12 é possível verificar a dinâmica da resposta da malha de controle conforme a configuração mostrada pela Fig. 7.9, bem como a dinâmica de cada estado observados e a entrada em porcentagem de potência. Os testes foram realizados para os *setpoints* de: 70, 120 e 170 °C.

Nos gráficos das figuras a seguir as curvas representam a resposta do sistema, o *setpoint* e os estados observados. As curvas são identificadas pela legenda apresentada a seguir:

Legenda:

- 1- Estado observado 1 (x_1 = temperatura);
- 2- Estado observado 2 (x_2);
- 3- Estado observado 3 (x_3);
- 4- Ponto de ajuste (*Setpoint*);
- 5- Corrente aplicada de 4 a 20 mA;
- 6- Temperatura controlada.

FIGURA 7.10 - Curvas de resposta do experimento do controle do sistema térmico
Setpoint de 70 °C

FIGURA 7.11 - Curvas de resposta do experimento do controle do sistema térmico
Setpoint de 120 °C

FIGURA 7.12 - Curva de resposta do experimento do controle do sistema térmico
Setpoint de 170 °C

7.3 RESULTADOS SIMULADOS

Os resultados dos testes obtidos através das estratégias utilizando os novos blocos funcionais desenvolvidos foram obtidos através de simulações no *software Matlab*, utilizando o sistema térmico real apresentado na Fig 7.1.

Os novos blocos funcionais desenvolvidos foram implementados com seus parâmetros de tal maneira a simular as situações o mais próximo do real.

São apresentados a seguir algumas estratégias de controle com objetivo de mostrar o comportamento do sistema de controle através da aplicação dos novos blocos funcionais implementados. Deve ser notado que os dados utilizados na simulação são os mesmos obtidos no item 7.2.

7.3.1 ESTRATÉGIA DE CONTROLE POR OBSERVAÇÃO DE ESTADOS

Para simular esta estratégia foram necessários, além do novo bloco Observador de Estados (OE), implementar os blocos padrões : Entrada Analógica (AI), Saída Analogia (AO), Constante (CT) e Aritmético. A Fig. 7.12 mostra a configuração da estratégia de controle, bem com a configuração do bloco aritmético e observado de estados:

FIGURA 7.13 - Estratégia de controle por observação de estados e a configuração dos parâmetros dos blocos

7.3.2 RESULTADOS OBTIDOS DA SIMULAÇÃO POR OBSERVAÇÃO DE ESTADOS

Nas Figs.(7.14, 7.15, 7.16, 7.17 e 7.18) é possível verificar a dinâmica da resposta da malha de controle simulada conforme a configuração mostrada pela Fig. (7.9), bem como a dinâmica de cada estado estimado e a entrada de referência. Os testes foram realizados para os *setpoints* de: 70, 120 e 170 °C.

Nos gráficos das figuras a seguir as curvas representam a resposta do sistema, o *setpoint* e os estados observados. As curvas são identificadas pela legenda apresentada a seguir:

Legenda:

1- Ponto de ajuste (<i>Setpoint</i>)	4- Estado observados 2 (x_2);
2- Temperatura controlada	5- Estado observados 3 (x_3);
3- Estado observado 1 (x_1 = temperatura);	

FIGURA 7.14 - Curva simulada de resposta no controle do sistema térmico
Setpoint de 70 °C

FIGURA 7.15 - Curva simulada de resposta no controle do sistema térmico
Setpoint de 120 °C

FIGURA 7.16 - Curva simulada de resposta no controle do sistema térmico
Setpoint de 170 °C

FIGURA 7.17 - Curva simulada de resposta no controle do sistema térmico
Setpoint de 70 e 140 °C

FIGURA 7.18 - Curva simulada de resposta no controle do sistema térmico
Setpoint de 100 e 70 °C

7.3.3 ESTRATÉGIA DE CONTROLE ADAPTATIVA

Para simular esta estratégia foram necessários os blocos Observador de Estados (OE), Adaptativo (AP) e os blocos padrões: Entrada Analógica (AI), Saída Analógica (AO) e Constante (CT). A Fig. 7.17 mostra a configuração da estratégia de controle, bem com a configuração dos blocos.

FIGURA 7.19 - Estratégia de controle adaptativa e a configuração dos parâmetros dos blocos

7.3.4 RESULTADOS OBTIDOS DA SIMULAÇÃO ADAPTATIVA

Na Fig.(7.18 e 7.20) é possível verificar a dinâmica da resposta da malha de controle simulada, conforme a configuração mostrada pela Fig. 7.17. Cabe se notar que a saída da planta acompanha a saída do modelo de referência. A Fig. (7.19 e 7.21) mostra o erro dado pela diferença do valor entre a curva do modelo e da planta.

FIGURA 7.20 - Curva de resposta do modelo e da planta para uma entrada degrau de 30%

FIGURA 7.21 - Erro para entrada degrau 30% (diferença saída do modelo e da planta)

FIGURA 7.22 - Curva de resposta do modelo e da planta para uma entrada degrau de 60%

FIGURA 7.23 - Erro para entrada degrau 60% (diferença saída do modelo e da planta)

FIGURA 7.24 - Curva de resposta do modelo e da planta para uma entrada degrau de 20, 40 e 60%

FIGURA 7.25 - Erro para entrada degrau 20, 40 e 60% (diferença saída do modelo e da planta)

A matriz P obtida para garantir a estabilidade do sistema foi obtida de acordo com a teoria *Lyapunov*, onde:

$$A_m^T P + P A_m = -Q$$

Logo:

$$P = \begin{bmatrix} 2,4983 \times 10^{-9} & 2,7197 \times 10^{-6} & 0 \\ 2,7197 \times 10^{-6} & 3,4262 \times 10^{-3} & 6,5989 \times 10^{-3} \\ 0 & 6,5989 \times 10^{-3} & 9,904 \times 10^0 \end{bmatrix}$$

Capítulo 8

Conclusão

Neste capítulo apresenta-se a conclusão do trabalho.

8.1 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de novos blocos funcionais para realizar o controle adaptativo por modelo de referência utilizando variáveis de estado em redes *Fieldbus*. As novas estratégias construídas trazem uma maior facilidade em controlar plantas com múltiplas entradas e múltiplas saídas em *Fieldbus*, devido ao fato de trabalhar com variáveis de estado. Isso demonstra um desenvolvimento de controle de processos com processamento distribuído em rede; caracterizando como uma aplicação essencialmente do protocolo *Fieldbus Foundation*.

Essas novas estratégias de controle realizam os cálculos de maneira distribuída entre os dispositivos da rede *Fieldbus*, implicando em várias vantagens, tais como: a) perda do supervisor não implica a perda do algoritmo de controle, b) menor tráfego de dados na rede, c) algoritmo de controle que não depende do tempo de processamento do computador, d) controlar sistemas que variam suas características dinâmicas ao longo do tempo ou faixa de operação. Além de ser uma alternativa para o controlador PID existente.

No entanto, existem também algumas desvantagens, sendo que a maior está relacionada à dificuldade para a identificação do modelo do processo e às condições iniciais no bloco adaptativo.

A nova estratégia de controle por estimação de estados foi testada utilizando uma configuração típica encontrada na prática, ilustrada na Fig. 7.1. A planta real constitui de um sistema de controle de temperatura instrumentada por dispositivos (sensores e transmissores) e interligados através da rede Fieldbus a um microcomputador do tipo PC para realizar a configuração e a supervisão da planta. Os resultados mostraram claramente que a estratégia proposta obteve um alto grau de precisão na estimação dos estados e controle da saída para uma variedade de condições, tais como: *Setpoint* de 70°C, 120°C e 170°C. O erro percentual em termos de exatidão na maioria dos casos analisados foi menor que 3 %.

A mesma estratégia de controle por realimentação foi também testada utilizando o *software Matlab*, onde foram implementados os blocos padrões e os novos blocos funcionais desenvolvidos. Os resultados obtidos mostraram um comportamento bastante semelhante dos valores encontrados na situação real, confirmando o bom desempenho da malha de controle e demonstrando o alto grau de fidelidade dos blocos simulados.

Por último, foi simulada a estratégia de controle adaptativa por modelo de referência, utilizando o *software Matlab* e os blocos implementados. As simulações mostraram que os resultados foram satisfatórios para diversas situações, tornando a malha de controle eficaz e confiável. O erro percentual em termos de exatidão na maioria dos casos analisados foi menor que 5 %. Porém a estratégia adaptativa, possui o inconveniente de necessitar de condições iniciais para começar o processo de iteração no controle da planta.

Finalmente, conclui-se que os testes realizados mostram a funcionalidade dos blocos funcionais e que as estratégias apresentadas demonstraram um alto grau de precisão para as diversas situações testadas, destacando-se sua confiabilidade e aplicabilidade, bem como a viabilidade de sua implementação prática.

8.1.1 TRABALHOS FUTUROS

Algumas sugestões para trabalhos futuros nesta linha de pesquisa:

- Projeto e implementação de novos blocos funcionais utilizando outras técnicas de controle avançado como: controle preditivo, redes neurais, lógica *fuzzy*, etc..;
- Desenvolvimento de um *software* para determinar automaticamente o modelo de referência dos blocos;
- Implementação de sistema de controle avançados, para aplicação de controle de plantas não-lineares.

Capítulo 9

Referências Bibliográficas

ÅSTRÖM, K. J.; WITTERNMARK, B. (1989) Adaptive control. Addison Wesley Publishing Company.

ÅSTRÖM, K. J.; WITTERNMARK, B. (1995) Adaptive control. 3.ed. New York, Addison-Wesley.

BRANCO, E. L. S. (1996). *Desenvolvimento de um Controlador Avançado Aplicado a Processos Industriais*. Dissertação (Mestrado), Escola Federal de Uberlândia, Uberlândia, Brasil.

BRANDÃO, D. (2000). *Bloco Funcional para controle Fieldbus por variáveis de estado*. Dissertação (Mestrado), Escola de Engenharia de São Carlos, Universidade de São Paulo, S.Carlos, S.Paulo, Brasil.

BLEVINS, L. T.; PELUSO, M. (1999). Advanced control capability of Fieldbus device. In: Conferência Internacional ISA SHOW 99.

- CAVALIERI, S.; DI-STEFANO, A.; MIRABELLA, O. (1996). Mapping automotive process control on IEC/ISA Fieldbus functionalities. *Computers in industry*. Vol.28, n.3, p.233-250, jun.
- CHALAN, V. V. (1987). Adaptive control systems. *Techniques and Applications*. Marcel Dekker, INC., New York.
- CHAO, C. L.; NEOU, J. (1999). Model reference adaptive control of air-lubricated capstan drive for precision positioning. *Journal of the International Societies for Precision Engineering and Nanotechnology*, v.35, p.285-290.
- DOMINGUES, E. T.; PINOTTI, M.; PACHECO, A. C. C. (2002). Proposta de uma nova estratégia em redes *Fieldbus* usando controle adaptativo por modelo de referência. II congresso nacional de engenharia mecânica, João Pessoa, Paraíba.
- DRWAL, A.; WEREWKA, J. (1999). Design of a Fieldbus network structure for distributed control systems. *Kwartalnik Elektroniki i Telekomunikacji*, v.45, n.1, p.51-72.
- DUARTE, A. M.; NARENDRA, K. S. (1989). Combined direct and indirect approach to adaptive control. *IEEE Transaction Automatic Control*, AC-34, p.1071-1075.
- DUARTE, A. M.; NARENDRA, K. S. (1996). Indirect model reference adaptive control with dynamic adjustment of parameters. *Internacional Journal of Adaptive and Control and Signal Processing*, v10, p.603-621.
- DULTRA, M. L.; VELONI, V. (1996). Boiler control with Fieldbus. *FieldComms UK. The Industrial Networking Show and Conference*. GGH Marketing Commun, Titchfield, UK; pp. 236, p.85-90.

- ENGELMANN, A. S.; CASTRO, D. J.; TORRES G. L.; FRANCO, L. R. H. R. (2001). Solving the scheduling problem of messages in fieldbus systems with a genetic algorithm. *In: Conferencia Internacional ISA SHOW 2001.*
- FERRAZ, W. D. (1999). Adaptive Control Using Fieldbus Resources. *In: Conferencia Internacional ISA SHOW 99.*
- FRANCO, L. R. H. R. (1998). *Escalonamento de mensagens para a comunicação no Fieldbus.* Tese (Doutorado), Escola Politécnica de São Paulo, S.Paulo, Brasil.
- FRANCO, L. R. H. R. (1999). *Protocolos de Comunicação que substituem as fiações elétricas de Veículos,* IME - Instituto Militar de Engenharia, Rio de Janeiro, dezembro de 1999.
- FRANKLIN, G.F. et al. (1994) Feedback control of dynamic systems. 3.ed. New York, Addison-Wesley.
- FIELDBUS FOUNDATION (1999). *Foundation Specification.* System Architecture, FF-800-1.4, Final Specification
- FIELDBUS FOUNDATION (1999). *Foundation Specification.* Function block Application Process –Parte- 1, FF-890-1.4, Final Specification
- FIELDBUS FOUNDATION (1999). *FoundationTM Specification.* Function block Application Process –Parte- 2, FF-891-1.4, Final Specification.
- FIELDBUS FOUNDATION (1999). *FoundationTM Specification.* Function block Application Process –Parte- 3, FF-892-1.4, Final Specification.

FIELD BUS FOUNDATION (1999). *FoundationTM Specification*. Function block Application Process –Parte- 4, FF-893-1.4, Final Specification.

FIELD BUS FOUNDATION (1999). *FoundationTM Specification*. Function block Application Process –Parte- 5, FF-894-1.4, Final Specification.

FIELD BUS FOUNDATION (1999). *Foundation Specification*. Network Management, FF-801-1.4, Final Specification

GLANZER, D. A.; SANTORI, M. L. (1998). PC-based control systems using Fieldbus. *National Manufacturing Week Conference Proceedings '98, 'Preparing Industry for the 21st Century'. Reed Exhibition, Norwalk, CT, USA; v.(v+216+232+188), p.61-66.*

HARROLD, D. (1999). Incomplete combustion burns money. *Control engineering international*, p.46-50, nov

HODSON, W. R. (1998). Exploiting non-control functions in field devices with Fieldbus. *TAPPI Proceedings. Joint Conference Process Control, Electrical and Information Conference ISA Pupid 38th Annual Symposium*. TAPPI Press, Atlanta, GA, USA, p.(vii+655).

KOJIMA, M.; IKEDA, T.; OSHIMA, T.; TANABE, A. (1997). *Fieldbus and intelligent devices for oil production plants. Offshore-Technology-Proceedings-of-the-International-Conference-on-Offshore-Mechanics-and-Arctic-Engineering-OMAE*. ASME, New York. V. 1.

- KREISSELMEIER, G.; SMITH, M. C. (1986). Stable adaptive regulation of arbitrary n-th order plants. *IEEE Transaction Automatic Control*, AC-31, p.289-305.
- LI, J.; WATANABE, A.; KAWATA, S. (1996). Local coefficients of Walsh functions and their applications to model reference adaptive control. *Proceedings of the 35th Conference on Decision & Control, Kobe, Japan*; p.3737-3742.
- MAHER, M.; DAHHOU, B.; ZENG, F. Y. (1995). Experimental results in model reference adaptive estimation and control of a fermentation process. *Control Eng. Practice*, V.3, n.3, p.313-320.
- MAHALIK, N. P.; MOORE P, R. (1997). Fieldbus technology based, distributed control in process industries: a case study with LonWorks technology. *Integrated Manufacturing Systems*. V.8, n.4, p.231-243.
- MAKOUDI, M.; RADOUANE, L. (2000). A robust model reference adaptive control of non-minimum phase systems with unknown or time-varying delay. *Automatica* 36, 1057-1065
- McLAIN, R. B.; HENSON, M. A. (2000). Nonlinear model reference adaptive control with embedded linear models. *Ind. Eng. Chem.* V.39, n.8, p.3007-3017, n. 8
- MIYASATO, M. (1998). A simple redesign of model reference adaptive control system and its robustness. *Proceedings of the 35th Conference on Decision & Control, Tampa, Florida, USA*; p.2280-2285.

MOCHIOKA, Y. (1999). Design concepts of Foundation/sup TM/ Fieldbus products. *Yokogawa-Technical-Report-(English-Edition)*. N.27, p.1-4, jun.

MURRILL, P. W. (1988). *Application Concepts of Process Control*. Instrument Society of America.

NAIM IDRA, K. S., ANNASWAMY A. M., (1989). Stable adaptive systems. Ed Prentice-Hall, 1989.

NONAKA, K.; YAMAKI TA, M.; FURUTA, K. (1996). A study on model reference variable structure control with adaptive sliding surface. *Proceedings of the 35th Conference on Decision & Control, Kobe, Japan*; p.3504-3509.

OGATA, K. (1997). *Engenharia de controle moderno*. Terceira edição, Ed Prentice-Hall, Rio de Janeiro, Brazil.

PHAN, M. Q.; FRUEH, J. A. (1999). Model reference adaptive learning control with basis functions. *Proceedings of the 38th Conference on Decision & Control, Phoenix, Arizona, USA*; p.251-257.

PACHECO, A. C. C. (1998). Estudo de algoritmos para escalonamento em rede Fieldbus. Dissertação (Mestrado), Escola de Engenharia de São Carlos, Universidade de São Paulo, S.Carlos, S.Paulo, Brasil.

PACHECO, A. C. C.; DOMINGUES, E. T.; PINOTTI, M. (2002). Proposta de novos blocos funcionais Fieldbus Foundation para realizar o controle adaptativo por realimentação de estados. II congresso nacional de engenharia mecânica, João Pessoa, Paraíba

- RAMACHANDRAN, R.; GOFF, D.; MANGANO, J.; DUMDIE, D. (1998). Distributed control using FOUNDATION Fieldbus on a hydrogen peroxide bleaching process. *TAPPI Proceedings. Joint Conference Process Control, Electrical and Information Conference ISA Pupid 38th Annual Symposium. TAPPI Press, Atlanta, GA, USA; pp.(vii+655), p.297-301.*
- RAGO, M. P. R.V. (1998). Non-quadratic Lyapunov and adaptive laws for model reference adaptive control. *Electronics Letters*, v.34, n.23, p.2278-2280, nov.
- SINHA, P. K.; HULME, A. J. (1883). Simulation and control of a non-linear electromagnetic suspension system. *In: J. Burger, & Y. Jarny (Eds.), Simulation in Engineering sciences. Amsterdam: Elsevier, p.257-262.*
- SINHA, P. K.; PECHEV, A. N. (1999). Model reference adaptive control of a magnetic levitation system with stable maximum descent criterion. *Automatica* 35, 1457-1467.
- SILVA, W. A. C. M.; ARAÚJO, A. D.; SALAZAR, A. O. (1997). Implementação de um controlador adaptativo com estrutura variável aplicado ao controle de velocidade de motores CC. *7th Congreso Latinoamericano en Control Automatico - Buenos Aires - Argentina - Setembro de 1996 - Vol.1 - pp. 2001-2002.*
- TSUNASHIMA, K.; HAO, S. H.; YANG, Z. J. (1996). Model reference nonlinear adaptive control with observer. *Proceedings of the 35th Conference on Decision & Control, Kobe, Japan; p.4681-4682.*
- TANENBAUM, A. S. (1996). *Computer network*. 3.ed. New Jersey, Prentice-Hall.
- VALLBOM, B. (1998). *Dinâmica e Controle de Processos*. Ed Tórculo artes Gráficas, S.A.L, Rosalía de Castro 45, Santiago, Chile, 172p.

- VALLERA, A.; SALT, J.; CASANOVA, V.; FERRUS, S. (1999). Control of industrial robot with a fieldbus. *7th IEEE International Conference on Emerging Technologies and Factory Automation. Proceedings ETFA 99 (Cat. No. 99TH8467). IEEE, Piscataway, NJ, USA; v.2, p.(xxiv+xviii+1573).*
- ZHANG, H. (1996). Fuzzy control with the neural network method in fieldbus control systems. *Mini-Micro-Systems*, v.17, n.11, p.60-64, nov.
- ZISSE, N.; SCHUMNY, H. (1997). Open communication with a fieldbus at petrol stations. *New-Frontiers-of-Information-Technology-Conference-Proceedings-of-the-EUROMICRO. 1997, IEEE Comp Soc, Los Alamitos, CA, USA. p 100-105*