UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E ENGENHARIA DE PETRÓLEO

Reconfiguração Dinâmica de Estratégias Distribuídas em Dispositivos *Foundation Fieldbus* para a Otimização de Processos na Indústria do Petróleo

Leonardo Sávio Guanabara Ramalho

Orientador: Prof. Dr. Adrião Duarte Dória Neto

Co-orientador: Prof. Dr. Jorge Dantas de Melo

Natal, Junho de 2009.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E ENGENHARIA DE PETRÓLEO

Reconfiguração Dinâmica de Estratégias Distribuídas em Dispositivos *Foundation Fieldbus* para a Otimização de Processos na Indústria do Petróleo

Leonardo Sávio Guanabara Ramalho

Orientador: Prof. Dr. Adrião Duarte Dória Neto

Co-orientador: Prof. Dr. Jorge Dantas de Melo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência e Engenharia de Petróleo da UFRN (área de concentração: Automação na Indústria de Petróleo e Gás Natural) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, Junho de 2009.

Reconfiguração Dinâmica de Estratégias Distribuídas em Dispositivos *Foundation Fieldbus* para a Otimização de Processos na Indústria do Petróleo

Leonardo Sávio Guanabara Ramalho

Dissertação de Mestrado aprovada em 29 de Junho de 2009 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Adrião Duarte Dória Neto (orientador) . . . . . DCA/UFRN

Prof. Dr. Jorge Dantas de Melo (co-orientador) . . . . . DCA/UFRN

Prof. Dr. Eduardo Oliveira Freire (examinador externo) . . . . . NEL/UFS

Prof. Dr. Luiz Affonso H. G. de Oliveira (examinador interno) . . . DCA/UFRN

Divisão de Serviços Técnicos
Catalogação da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Ramalho, Leonardo Sávio Guanabara.
Reconfiguração dinâmica de estratégias distribuídas em dispositivos *Foundation Fieldbus* para a otimização de processos na indústria do petróleo / Leonardo Sávio Guanabara Ramalho. – Natal, RN, 2009.
71 f. : il.

Orientador: Adrião Duarte Dória Neto.
Co-orientador: Jorge Dantas de Melo.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Ciência e Engenharia de Petróleo.

1. Redes neurais – Dissertação. 2. Petróleo – Indústria – Dissertação. 3. Sistemas inteligentes – Dissertação. 4. Foundation Fieldbus – Dissertação. I. Dória Neto, Adrião Duarte. II. Melo, Jorge Dantas de. III. Universidade Federal do Rio Grande do Norte. IV. Título.

RN/UF/BCZM CDU 004.032.26 (043.2)

Dedico esta dissertação de mestrado
à minha mãe Dulce Lêda
Guanabara que, ao longo de alguns
anos, vem me dando as lições mais
importantes para a minha formação.

Agradecimentos

Agradeço à minha família e à minha namorada Dani, pelo apoio e carinho constante durante esta jornada.

À Instituição UFRN e aos membros do Departamento de Engenharia de Computação e Automação, assim como aos membros do Programa de Pós-graduação em Ciência e Engenharia de Petróleo.

Aos professores e amigos, Adrião, Jorge e Affonso, que contribuíram para minha formação acadêmica e pessoal, além de confiarem sempre em meu trabalho.

Aos companheiros de trabalho e colegas que passaram pelo Laboratório de Avaliação de Medição em Petróleo da UFRN, pela grande ajuda no desenvolvimento do trabalho. Em especial, aos amigos Daniel Lopes, Bruno Costa e Vinícius Machado.

Resumo

A indústria petroquímica tem por objetivo obter, a partir do petróleo bruto, produtos de alto valor comercial e de maior utilidade industrial para fins energéticos. Os processos dessa indústria são complexos, geralmente operam com grandes volumes de produção e em condições restritas de operação. O controle da operação em condições ótimas e estáveis é importante para manter a qualidade dos produtos obtidos e a segurança da planta.

Atualmente, as redes industriais têm alcançado destaque quando há a necessidade de se realizar o controle do processo de forma distribuída. O protocolo para redes industriais *Foundation Fieldbus*, por sua característica de interoperabilidade e sua interface com usuário organizada em blocos de simples configuração, tem grande notoriedade dentre o grupo de redes para automação industrial.

O presente trabalho agrega os benefícios trazidos pela tecnologia de redes industriais à complexidade inerente dos processos ligados a industria petroquímica. Para tal, é proposto um sistema para reconfiguração dinâmica de estratégias inteligentes (redes neurais artificiais, por exemplo) baseado na camada de aplicação do usuário do protocolo, o qual poderá permitir o uso de diferentes aplicações em um determinado processo, sem intervenções de operadores e com as garantias necessárias para o bom funcionamento da planta.

Palavras-chave: Redes Industriais, *Foundation Fieldbus*, Sistemas Inteligentes, Redes Neurais.

Abstract

The petrochemical industry has as objective obtain, from crude oil, some products with a higher commercial value and a bigger industrial utility for energy purposes. These industrial processes are complex, commonly operating with large production volume and in restricted operation conditions. The operation control in optimized and stable conditions is important to keep obtained products quality and the industrial plant safety.

Currently, industrial network has been attained evidence when there is a need to make the process control in a distributed way. The *Foundation Fieldbus* protocol for industrial network, for its interoperability feature and its user interface organized in simple configuration blocks, has great notoriety among industrial automation network group.

This present work puts together some benefits brought by industrial network technology to petrochemical industrial processes inherent complexity. For this, a dynamic reconfiguration system for intelligent strategies (artificial neural networks, for example) based on the protocol user application layer is proposed which might allow different applications use in a particular process, without operators intervention and with necessary guarantees for the proper plant functioning.

Keywords: Industrial Network, *Foundation Fieldbus*, Intelligent Systems, Neural Networks.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos e Abreviaturas

AI: Analog Input

AO: Analog Output

BT: Background Traffic

CD: Compel Data

CNPQ: Conselho Nacional de Desenvolvimento Científico e Tecnológico

DCA: Departamento de Engenharia de Computação e Automação

DCS: Distributed Control System

DD: Device Description

DDC: Direct Digital Control

DDL: Device Description Language

DDT: Distributed Data Transfer

DI: Discrete Input

DLL: Data Link Layer

DO: Discrete Output

DSP: Digital Signal Processor

FAS: Fieldbus Access Sublayer

FCS: Field Control System

FF: Foundation Fieldbus

FINEP: Financiadora de Estudos e Projetos

FMS: Fieldbus Message Specification

FT: Foreground Traffic

IEC: International Engineering Consortium

ISA: International Society of Automation

ISO: International Organization for Standardization

ISP: InterOperable Systems Project

LAMP: Laboratório de Avaliação de Medição em Petróleo

LAS: Link Active Scheduler

MC: Macro-ciclo de Execução

MLP: Multi-Layer Perceptron

OLE: Object Linking Embedded

OPC: OLE for Process Control

OSI: Open System Interconnection

PCI: Peripheral Component Interconnect

PID: Controlador Proporcional-Integral-Derivativo

PLC: Programmable Logic Controllers

PN: Probe Node

PR: Probe Response

PT: Pass Token

RNA: Rede Neural Artificial

TD: Time Distribution

UFRN: Universidade Federal do Rio Grande do Norte

VCR: Virtual Communication Relationships

VFD: Virtual Field Device

XML: Extensible Markup Language

CAPÍTULO 1

Introdução Geral

1.1 Introdução

A evolução dos sistemas de controle tem sido bastante acentuada nas últimas décadas. Nas suas primeiras versões, os controladores atuavam através de sinais pneumáticos e eram localizados no campo, sendo operados localmente.

Posteriormente, com o controle feito através de sinais analógicos ou digitais, tornou-se possível levar o sinal para os transmissores no campo, caracterizando um Controle Direto Centralizado (*Direct Digital Control* - DDC), em que o dispositivo controlador (computador digital) se encontrava numa sala de controle específica. Segundo [Zerbetto 2007], esse tipo de arquitetura surgiu em 1962.

A arquitetura acima citada possuía uma fragilidade. No caso, o fato de toda organização do sistema ser concentrada em um único dispositivo fazia com que uma falha no mesmo gerasse uma parada total no sistema. Logo, em 1976 foi elaborado um novo tipo de arquitetura conhecida como Sistema de Controle Distribuído (*Distributed Control System* - DCS). As tarefas de controle passaram a ser distribuídas em diversos dispositivos, fazendo com que um falha afetasse apenas uma parte do processo. Tal avanço só foi possível com a criação dos Controladores Lógicos Programáveis (*Programmable Logic Controllers* - PLC), unidades independentes capazes de realizar processamento, concentrando atividades de controle e fornecendo informações sobre o processo [Zerbetto 2007].

Entretanto, atualmente, o DCS só é considerado menos centralizado que o DDC, visto que uma falha no sistema pode ocasionar outras falhas em cadeia [Berge 2001].

Uma nova arquitetura baseada numa maior capacidade dos dispositivos de campo é a alternativa mais proveitosa, a chamada arquitetura de sistemas de controle de campo (*Field Control Systems* - FCS) [Pantoni 2006].

Tal arquitetura consiste em uma rede industrial em que os sensores e atuadores do sistema compartilham um meio físico para transmissão da informação. Sua principal proposta é a distribuição do controle entre os dispositivos sensores, atuadores e controladores interligados em rede. Atualmente, diversos protocolos, os quais implementam a arquitetura FCS, encontram-se disponíveis no mercado (*Foundation Fieldbus*, *Profibus*, *DeviceNet*, *WorldFip*, por exemplo), cada qual com características próprias que os diferenciam, geralmente envolvendo o tipo de escalonamento de tarefas, políticas de acesso ao meio e composição dos *frames* de mensagens.

A figura 1.1 apresenta a evolução das arquiteturas dos sistemas de controle previamente comentada.

Dentre as várias tecnologias de Sistemas de Controle de Campo, destaca-se o protocolo *Foundation Fieldbus* (FF) por apresentar duas características particulares. Uma delas

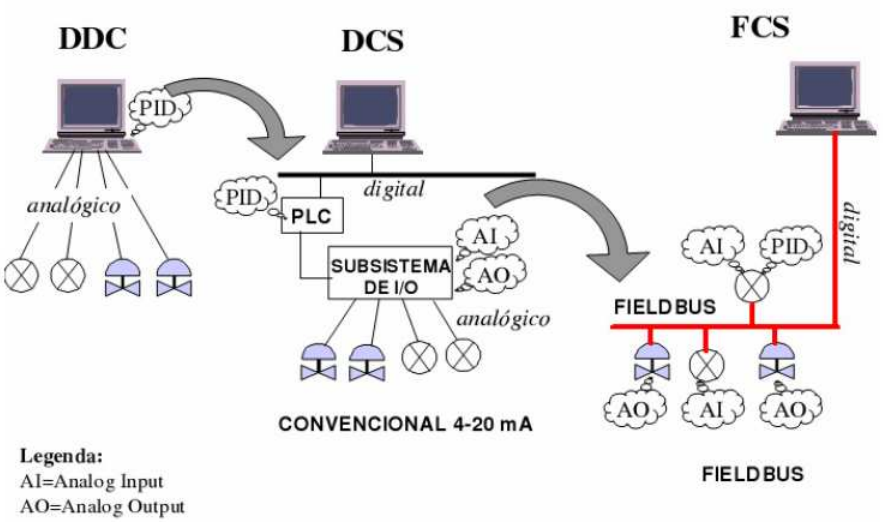

Figura 1.1: Evolução das Arquiteturas de Controle

é a padronização da camada de aplicação, que é dividida em blocos contendo funcionalidades específicas. E a segunda, como consequência da primeira somada à possibilidade de processamento distribuído, é a flexibilidade que a arquitetura permite ao engenheiro de automação para desenvolver sua estratégia de controle. Já que os dispositivos FF são providos de processador e memória e são capazes de executar seus próprios algoritmos de controle [Silva 2005].

A padronização da tecnologia garante a interoperabilidade entre os produtos das várias empresas associadas ao padrão, além de abrir a possibilidade de escolha, por parte dos clientes, dentre dispositivos de vários fabricantes.

Os blocos funcionais, correspondentes à camada de aplicação na tecnologia FF, têm um papel de importância no sucesso da interoperabilidade, já que os diferentes fabricantes seguem as especificações de implementação dos algoritmos internos dos blocos. Essa solução simplifica o desenvolvimento de diversas funcionalidades por permitir implementações independentes, as quais apresentam uma *interface* bem definida com o resto do ambiente, viabilizando uma fácil integração de novos algoritmos aos já existentes.

1.2 Escopo do Trabalho

O grande interesse no paradigma de redes industriais se deve ao potencial de possíveis novas aplicações e funcionalidades na área de instrumentação e controle, tais como: medição indireta de variáveis de processo, detecção de falhas e degradação em sistemas

de medição e implementação distribuída de estratégias de controle de processos, via concepção e implementação de sensores de *software*.

Tais aplicações tornam-se viáveis a partir da integração da tecnologia das redes industriais com algum tipo de estrutura inteligente capaz de servir de suporte à implementação das funcionalidades comentadas acima. No caso, as características previamente citadas do protocolo FF tornam possível sua integração com estruturas como as Redes Neurais Artificiais (RNA) ou Lógica *Fuzzy*, as quais podem atuar como ferramentas inteligentes para a concepção dos sensores de *software*.

Os sensores virtuais ou sensores de *software* são algoritmos computacionais utilizados para resolver problemas como inferência de variáveis, predição para controle de uma planta, estratégias de diagnósticos de falhas, dentre outros [Fortuna et al. 2006].

Um exemplo bastante utilizado no meio industrial é o da medição indireta. O sensor de *software* é empregado no lugar de um sensor físico para obter a medição de uma variável do processo a partir de valores de outras variáveis conhecidas [Fortuna et al. 2005] [Zanata 2005].

A alternativa para reconfiguração dinâmica das estratégias distribuídas nos instrumentos da rede industrial se integra justamente na tentativa de dar maior adaptabilidade ao protocolo de rede em questão. Com a possibilidade de se realizar reconfiguração de blocos funcionais em execução, torna-se viável a implementação de diferentes sensores de *software* agregados a uma rede industrial, os quais serão acionados sem intervenção na planta do processo.

1.2.1 Possíveis Aplicações

Dentre alguns trabalhos que tratam de sistemas inteligentes aplicados à processos da indústria petroquímica, há o exemplo demonstrado em [Zanata 2005], no qual realiza-se a criação de um sensor virtual aplicado a uma coluna de destilação que é capaz de estimar a composição dos produtos no topo da coluna utilizando, para tal, redes neurais artificiais. Em [Fortuna et al. 2005], são apresentados alguns sensores de *software* aplicados a diferentes processos de uma refinaria, com o intuito de diminuir custos operacionais e diminuir danos ao meio ambiente, a partir de informações coletadas em tempo real.

As aplicações desenvolvidas nos trabalhos citados acima baseiam-se na mesma ferramenta computacional para a obtenção de seus resultados. Devido a flexibilidade existente na camada de aplicação do usuário do protocolo FF, faz-se possível o uso da ferramenta rede neural implementada através de blocos funcionais padronizados pela tecnologia [Silva 2005]. A metodologia para reconfiguração dos algoritmos instanciados na rede,

por sua vez, torna viável a integração das diferentes funcionalidades em um determinado processo.

Outros exemplos de aplicações de diferentes funcionalidades podem ser vistos em [Cagni et al. 2005], em que há a realização da implementação de algoritmos para auto-calibração, auto-compensação e auto-validação de sensores *Fieldbus*. Assim como em [Costa 2006], em que algoritmos para filtragem estocástica são implementados nos sensores de uma rede industrial *Foundation Fieldbus*.

Uma outra abordagem para as aplicações consiste na implementação de agentes inteligentes distribuídos no barramento de campo da rede industrial. Em [Buse & Wu 2004] e em [Brennan et al. 2002], por exemplo, são apresentadas abordagens voltadas para teoria de agentes, as quais estão associadas a sistemas de controle distribuído.

Um dos objetivos associados ao sistema de reconfiguração das estratégias de controle distribuído nos dispositivos da rede consiste na tentativa de prover uma maior autonomia à rede industrial, tornando o sistema como um todo capaz de realizar diagnóstico sobre o funcionamento da operação e executar determinadas tarefas de correção para possíveis falhas.

A arquitetura multi-agente aplicada a uma rede industrial FF é proposta em [Machado et al. 2008*a*] [Machado et al. 2008*b*]. Tal arquitetura baseia-se na presença de agentes inteligentes no barramento de campo de uma rede industrial realizando diferentes tarefas de acordo com as necessidades do sistema (processo + rede industrial).

A alternância entre a ativação dos diferentes tipos de agente possui como base de implementação a reconfiguração da rede industrial.

1.3 Motivação e Objetivos do Trabalho

A idéia para concepção deste trabalho surgiu das atividades desenvolvidas em dois projetos de pesquisa realizados pelo grupo de Sistemas Inteligentes do DCA da UFRN.

O primeiro deles, o REDIC_SEN, criado através de um financiamento em conjunto entre CENPES e FINEP, visava o desenvolvimento de sensores de *software* em ambiente de redes industriais aplicados aos problemas de medição e controle na indústria do petróleo.

O projeto gerou contribuições no aumento da eficiência e custo operacional dos processos, e na disseminação do uso da tecnologia de sensores inteligentes.

O principal produto desenvolvido pelo projeto foi um sistema de concepção e implementação de redes neurais em sistemas supervisórios e/ou dispositivos de campo interligados em rede FF. A partir desse sistema, foram testadas as possibilidades de se implementar

novas funcionalidades para medição indireta, detecção de falhas, controle de processos e modelos de inferência.

O segundo projeto de pesquisa, intitulado "Arquitetura Multi-Agente Baseada em Blocos Funcionais em Redes Industriais *Foundation Fieldbus*", foi criado através de um financiamento do CNPq e teve como principal objetivo a inclusão de agentes inteligentes [Machado et al. 2008*a*] [Machado et al. 2008*b*] no barramento de campo de uma rede industrial.

Logo, utilizou-se a estrutura criada para a concepção dos sensores de *software* do primeiro projeto para a implementação de um sistema de reestruturação em tempo de execução das estratégias distribuídas nos dispositivos da rede FF, com o objetivo de auxiliar a implementação dos agentes inteligentes implementados no segundo projeto.

O objetivo do trabalho consiste, portanto, no estabelecimento de uma metodologia que possa prover a reconfiguração dinâmica das estratégias distribuídas em uma rede *Foundation Fieldbus*, trazendo um ganho relativo no âmbito dos sistemas para automação industrial, já que características, como flexibilidade e adaptabilidade da rede, serão incrementadas com o sistema proposto.

Serão apresentados diferentes tipos de aplicações para sensores de *software* e a maior contribuição da proposta em questão é a possibilidade de usufruir dos diferentes tipos de algoritmo de forma automática, contribuindo com características como flexibilidade e adaptabilidade da rede industrial. Para tal, serão discutidos tópicos essenciais como a segurança do sistema e o comportamento do processamento distribuído para alterações de estratégias alocadas nos dispositivos da rede.

Um dos exemplos práticos para validação será a alteração entre duas estratégias de controle inteligente para uma planta especificada no anexo A. No caso, serão discutidos tópicos como a questão de intertravamento durante a alteração das estratégias, assim como tempo de resposta do sistema para uma nova configuração dos blocos pré-instanciados na rede industrial.

1.4 Estrutura da Dissertação

O presente documento está estruturado da seguinte forma: Os capítulos 2 e 3 apresentam toda a base teórica do trabalho. São abordados tópicos específicos sobre o protocolo de redes industriais *Foundation Fieldbus* e algumas características das Redes Neurais Artificiais.

O capítulo 4 demonstra a metodologia utilizada para a concepção do trabalho. No caso, o modo como a proposta foi desenvolvida, destacando material e tecnologias uti-

lizadas.

Já o capítulo 5, relata os testes realizados com os respectivos resultados. Por fim, no capítulo 6 são realizadas as considerações finais, com os comentários sobre dificuldades encontradas e a contribuição alcançada com o trabalho.

CAPÍTULO 2

Protocolo Foundation Fieldbus

2.1 Introdução

O caminho para a tecnologia de rede de campo teve seu início nos anos 70 com as primeiras tentativas de realizar controle distribuído no nível de dispositivos [Foundation n.d.]. Com a introdução dos Sistemas de Controle Distribuído (DCS), tornou-se possível espalhar o controle de forma inteligente através das instalações do processo.

No entanto, o nível de comunicação entre sensores e atuadores era praticamente inexistente, assim como a quantidade de dados enviada até o elemento central do DCS. Grande parte dos dispositivos se comunicava com os controladores através de sinais pneumáticos ou sinais analógicos (baseados em *loop* de corrente de 4-20mA). Informações reais do processo eram limitadas e geralmente obtidas através de interpolações, inferências e tecnologias proprietárias (sistemas de aquisição de dados ou *gateways* simplificados) de alto valor comercial. Os níveis de complexidade e custo eram elevados e os sistemas possuíam sérias restrições em termos de acesso a informação em tempo real.

Na década de 80, esforços consideráveis tornaram possível um elevado grau de desenvolvimento de um padrão para comunicação digital entre dispositivos de campo. Os responsáveis por tal evolução formavam o comitê ISA SP50, o qual passou alguns anos definindo detalhes técnicos e construindo um consenso para um padrão digital de rede de campo.

De maneira equivalente, os maiores fabricantes na área de controle de processos continuavam a investir em pesquisas, buscando a criação de protocolos de comunicação digital. Esses múltiplos esforços culminaram em um punhado de protocolos, os quais não possuíam forma de interconexão [Foundation n.d.].

No fim do ano de 1994, a tecnologia para rede de campo ganhou uma direção promissora. Dois consórcios paralelos, ISP (InterOperable Systems Project) e o WorldFIP, se fundiram e formaram a *Fieldbus Foundation*. Essa nova organização trouxe, na tentativa de consolidar um padrão internacional, um conjunto de esforços críticos de maneira rápida, como programas desenvolvidos pela organização, ensaios de campo e o estabelecimento de programas rigorosos para teste e registro de dispositivos *fieldbus* [Foundation n.d.].

A *Fieldbus Foundation* é uma organização independente que não visa lucro e cujo propósito é desenvolver e manter um padrão uniforme de redes de campo para automação de processos: o *Foundation Fieldbus*. Os membros incluem usuários e fabricantes dos dispositivos de campo e dos sistemas de automação. O objetivo da *Fieldbus Foundation* foi, e ainda é, criar um padrão simples, internacional, de redes de campo para áreas classificadas e que foi amplamente difundido como rede de campo padronizada pelo IEC

(International Engineering Consortium).

2.2 Características Gerais do Protocolo

Estruturalmente, o *Foundation Fieldbus* é um protocolo de comunicação determinístico, bidirecional, digital e multiponto inspirado no modelo de referência ISO/OSI de sete camadas para protocolos de comunicação digitais. O protocolo apresenta uma série de características que o tornam um padrão interessante para o uso em aplicações industriais, das quais podem ser citadas:

- segurança intrínseca para áreas de risco;
- barramento com alimentação para os sensores/atuadores;
- topologia em linha ou em árvore;
- capacidade de comunicação multi-mestre;
- comportamento dinâmico determinístico;
- transferência de dados distribuída (DDT);
- modelos de blocos padronizado para "interfaceamento" uniforme dos dispositivos;
- opções de extensões flexíveis baseadas em descritores de dispositivos (DD) [Sam 2000].

Tais características são embasadas em um conjunto de normas, o qual torna o padrão apropriado para o emprego em aplicações industriais [Berge 2001].

Para a descrição das funções de um dispositivo e para a definição de um acesso uniforme aos dados, o *Foundation Fieldbus* contém blocos de funções pré-definidos. Os blocos funcionais implementados em um dispositivo fornecem informações sobre as tarefas que o dispositivo pode executar. A tabela 2.1 apresenta os principais blocos de função (alguns blocos terão seu funcionamento detalhado no capítulo 4).

Tabela 2.1: Lista de Blocos Funcionais Padrões

AI	Entrada Analógica
AO	Saída Analógica
DI	Entrada Discreta
DO	Saída Discreta
PID	Controle PID
ARITH	Bloco Aritmético
CHAR	Bloco Caracterizador

A mudança para o paradigma de automação de processos em rede - desde o nível da automação até o nível de campo - resulta em uma abordagem flexível e com processamento distribuído. Isso diminui a carga em uma estação de controle de processos centralizada, a qual pode ser substituída inteiramente por instalações de pequena escala. Assim, uma malha de controle inteira pode ser implementada com pequenas unidades, consistindo apenas de um sensor e um atuador com um controlador de processos integrado se comunicando através da rede FF, como pode ser visto na figura 2.1 [Silva 2005].

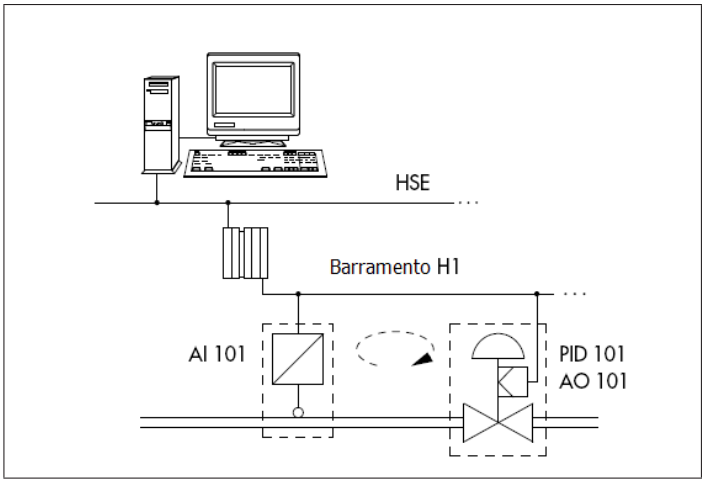

Figura 2.1: Malha de Controle baseada na tecnologia FF

2.3 Modelo de Comunicação em Camadas

A especificação *Foundation Fieldbus* é baseada em um modelo de comunicação em camadas, consistindo de três elementos principais (figura 2.2a). O nível físico FF equivale ao da camada física OSI. Sua principal particularidade é o fato de que a alimentação dos dispositivos pode usar o mesmo cabo em que trafegam os dados. A camada de comunicação é composta por três subcamadas: a subcamada inferior de enlace de dados (responsável pelo acesso ao meio físico e controle de erro); a subcamada intermediária de acesso a serviços da rede - *Fieldbus Access Sublayer* (FAS) - e a subcamada superior de montagem de mensagens - *Fieldbus Message Specification* (FMS) [Brandão 2005]. A camada de aplicação do usuário garante a interoperabilidade entre os dispositivos, sendo padronizada e completamente especificada. O protocolo não usa os níveis OSI 3, 4, 5 e 6. Cada nível no sistema de comunicação é responsável por uma parte da mensagem que é transmitida no barramento. A seguir, serão detalhados todos os níveis presentes no protocolo FF.

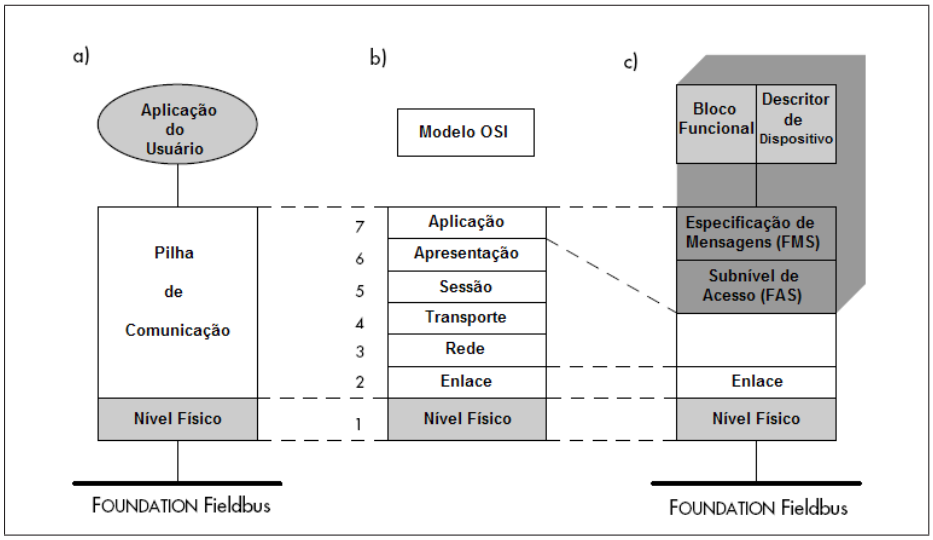

Figura 2.2: Modelo de Comunicação em Camadas

2.3.1 Nível Físico

A camada física é responsável por transformar os dados recebidos da camada de enlace de dados em sinais digitais que irão trafegar pela rede industrial, assim como realizar o "interfaceamento" para o sentido contrário. Essa camada também é responsável por inserir e remover o preâmbulo e o delimitador de início e fim de mensagem.

Os dados são codificados utilizando a codificação *Manchester Bifase L*, formando um sinal conhecido como "série sincronizada", pois codifica no sinal a informação do relógio. Utilizando esta codificação, o receptor dos dados interpreta uma transição positiva, que ocorrer no meio do tempo de transmissão de um *bit*, como sendo um sinal lógico "0" e uma transição negativa como sendo um sinal lógico "1" [Foundation n.d.] [Fie 2003].

O preâmbulo é utilizado para sincronização do relógio interno do receptor com o da mensagem para poder decodificá-la corretamente. O delimitador de início e de fim de quadro utilizam caracteres especiais N+ e N- que são transições de sinal, as quais não ocorrem no meio do *bit* como os dados normais. Quando o receptor encontra um delimitador de início de mensagem ele passa a receber dados até encontrar um delimitador de fim de mensagem. Nas figuras 2.3 e 2.4 são exibidos a codificação dos dados, do preâmbulo e dos delimitadores de início e fim de mensagem do protocolo FF.

O dispositivo que está se comunicando gera uma corrente de $\pm$ 10 mA numa taxa de 31,25 kbits/s que passa por uma carga de 50 ohm gerando uma variação de tensão de 1V de pico a pico, sendo esta corrente modulada sobre a tensão da fonte de alimentação, podendo esta última variar entre 9 e 32V. No fim e no início do cabo são utilizados terminadores de

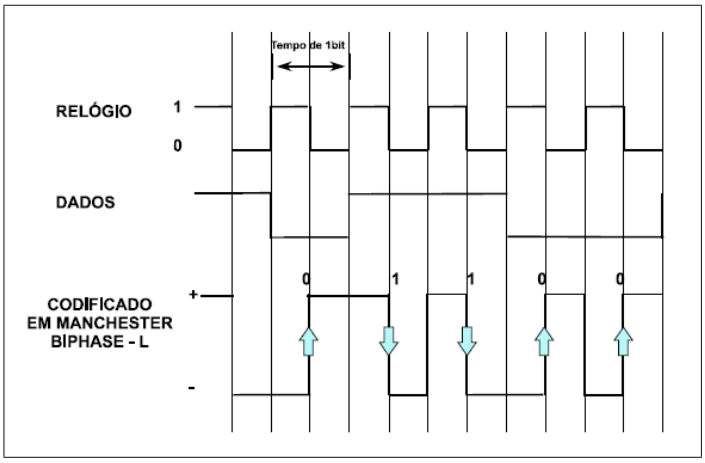

Figura 2.3: Codificação Manchester Biphase-L

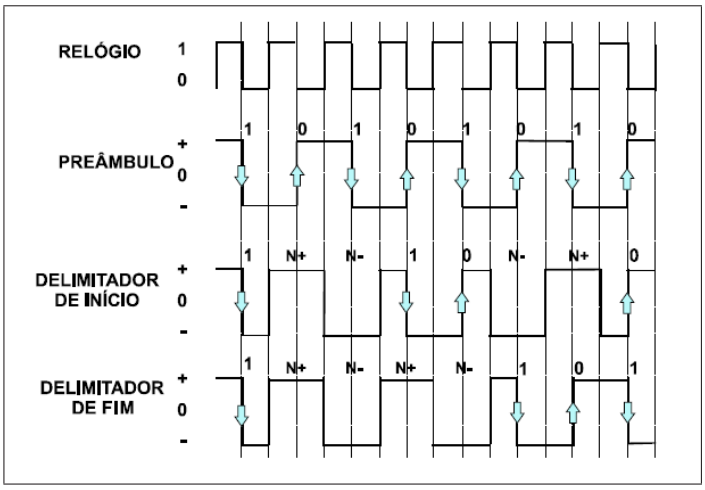

Figura 2.4: Preâmbulo, delimitador de início e fim do quadro FF

barramento que são filtros passa-faixa compostos por circuitos RC (resistor e capacitor) e que são projetados para passar a freqüência de 31,25 kbits/s.

A figura 2.5 apresenta a modulação realizada neste barramento, conhecido no protocolo como H1.

2.3.2 Pilha de Comunicação (Enlace + Camada de Aplicação)

Como citado anteriormente, a pilha de comunicação é composta por três subcamadas: a subcamada inferior de enlace de dados, a subcamada intermediária de acesso a serviços da rede (FAS) e a subcamada superior de montagem de mensagens (FMS). As subcamadas FAS e FMS juntas formam a camada de aplicação da pilha de comunicação.

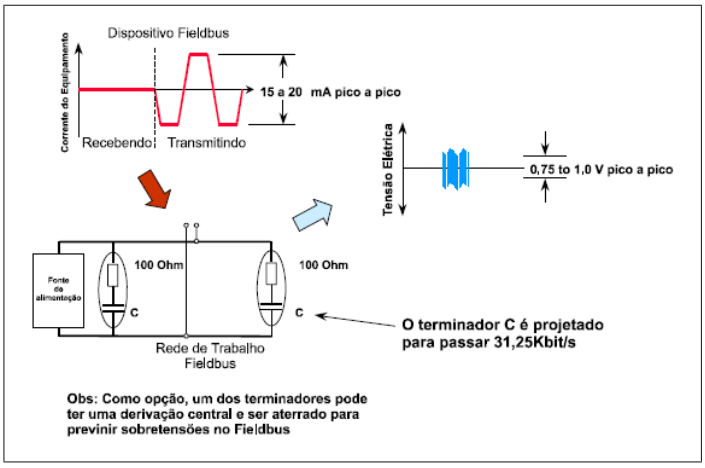

Figura 2.5: Sinalização no Barramento H1

A camada de enlace de dados, também conhecida como DLL (*Data Link Layer*), possui como principais características a responsabilidade de gerenciar políticas de acesso ao meio físico, uma vez que a topologia utilizada é de barramento, assim como, definir alguns dos primeiros procedimentos de tolerância à falha do sistema, baseando-se na abertura e fechamento de conexões, fluxo de controle e gerenciamento de erro [Cicillini 2007].

Os dispositivos de campo que seguem o protocolo *Fieldbus* são capazes de assumir funções relativas ao controle de processos. Essa opção baseia-se na comunicação distribuída, garantindo que:

- cada dispositivo de campo pode trocar informação com os demais dispositivos (como repassar a leitura de uma variável ou um valor corrigido);
- malhas de controle distintas não possuem interferência em suas execuções;
- dois ou mais dispositivos nunca acessam o barramento ao mesmo tempo.

Para tal, faz-se necessária a presença de um elemento central de comunicação: o *Link Active Scheduler* (LAS).

O LAS é um serviço de comunicação oferecido pela subcamada de DLL, o qual controla e escalona toda a comunicação sobre o barramento. É responsável pela organização das atividades no meio de comunicação, usando diferentes comandos, os quais são enviados a todos os dispositivos através de mensagens em modo *broadcast*. Uma de suas funções é a busca por novos dispositivos no barramento. Tal tarefa é realizada continuamente (é possível agregar novos dispositivos ao barramento em tempo de execução do processo).

Os dispositivos que são capazes de assumir tais funções são conhecidos como *Link Master*, caso contrário, recebem a denominação de *Basic Devices* [Berge 2001].

No caso de um sistema redundante, em que há mais de um instrumento *Link Master*, um deles poderá assumir o papel de LAS, se houver uma falha no LAS original.

O controle dos serviços de comunicação utiliza transmissões de dados síncronas e assíncronas. Tarefas de tempo crítico, como o controle de variáveis do processo, são exclusivamente realizadas por transmissões síncronas. Para o caso de tarefas como parametrização e funções de diagnótico, são utilizadas transmissões assíncronas [Sam 2000].

O conjunto das tarefas síncronas e assíncronas configura o macro-ciclo de execução da rede industrial.

Transmissão de Dados Síncrona

Todas as operações de tempo crítico são baseadas em transmissões de dados agendadas (síncronas). O escalonamento de tais tarefas deve ser criado pelo operador no momento de configuração do sistema.

LAS periodicamente envia (através de mensagens em modo *broadcast*) sinais de sincronização (TD: *Time Distribution*) no barramento para que os instrumentos possuam a referência em termos de tempo de execução [Sam 2000].

A tabela 2.2 apresenta um exemplo do escalonamento realizado para um sistema com dois sensores e duas válvula de controle. O agendamento realizado determina o momento de execução de cada função associada aos dispositivos (entrada analógica, saída analógica e controle PID).

Tabela 2.2: Escalonamento para Transmissão Síncrona

Dispositivo	Tipo	Ação	*Offset*
1	Sensor	execução AI(1)	0
		transmissão AI(1)	20
2	Sensor	execução AI(2)	0
		transmissão AI(2)	30
3	Atuador	execução PID(3)	40
		execução AO(3)	62
4	Atuador	execução PID(4)	40
		execução AO(4)	57

Cada atividade a ser executada é agendada para um certo tempo. Esse tempo é definido por um valor de *offset*, o qual reflete o atraso em relação ao começo do ciclo de execução.

Baseado nesse escalonamento, uma lista de transmissão é criada, a qual define quando um dispositivo específico está apto a enviar seu dado. No momento em que recebe a

mensagem, o respectivo instrumento (*publisher*) envia o dado em modo *broadcast* para todos os dispositivos do barramento que estão configurados para recebê-lo (*subscriber*).

O LAS realiza transmissões ciclicamente de acordo com a lista de dispositivos. Cada tarefa é ativada pela ação do LAS (figura 2.6), como nos exemplos abaixo:

- se um dispositivo está apto a "publicar" sua variável medida (o item "a" da figura 2.6), o LAS envia um comando CD (*Compel Data*) para o instrumento;
- diante do recebimento do comando CD, o dispositivo envia o valor para o barramento;
- os *subscribers* da mensagem (os ítens "b" e "c" da figura 2.6) podem ler e utilizar tal valor;

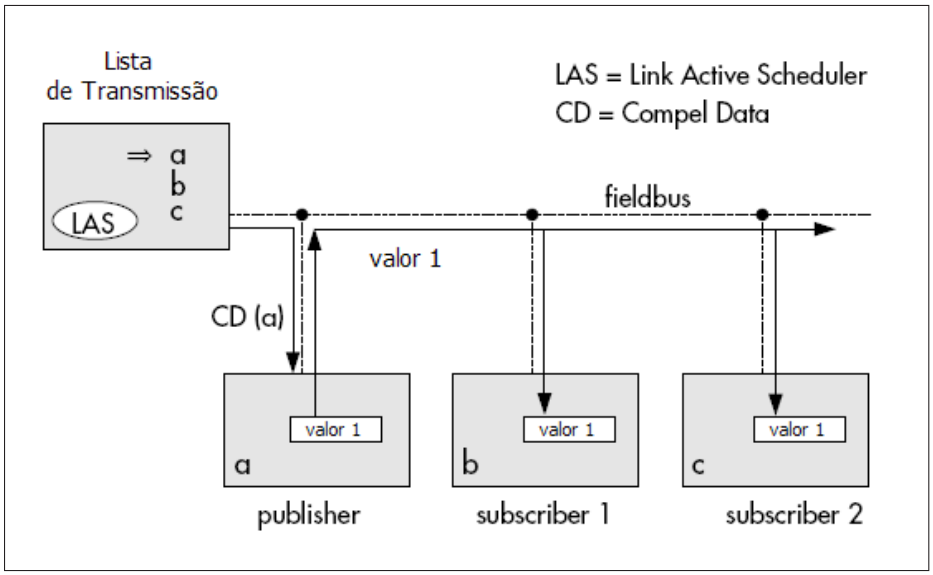

Figura 2.6: Escalonamento de acordo com Lista de Transmissão

Cada dispositivo de campo recebe um agendamento específico, tornando o gerenciamento mais simples. Cada tarefa tem um momento exato para acontecer, como o envio e a leitura de valores por um instrumento [Berge 2001].

A figura 2.7 ilustra a seqüencia de ações para um escalonamento realizado para as atividades da tabela 2.2. A seqüencia de passos é a seguinte:

- no tempo zero, os sensores 1 e 2 iniciam seus processos de medição;
- no tempo 20, o LAS permite ao sensor 1 o envio de sua medição para que possa ser lido pelo controlador PID associado a válvula de controle 3;

- no tempo 30, o LAS permite ao sensor 2 o envio de sua medição para que possa ser lido pelo controlador PID associado a válvula de controle 4;
- no tempo 40, as duas válvulas de controle estão processando suas funções;
- no tempo 57, a válvula 4 inicia sua atuação;
- no tempo 60, a válvula 3 inicia sua atuação;
- no instante de tempo 140, o processo reinicia (macro-ciclo de execução).

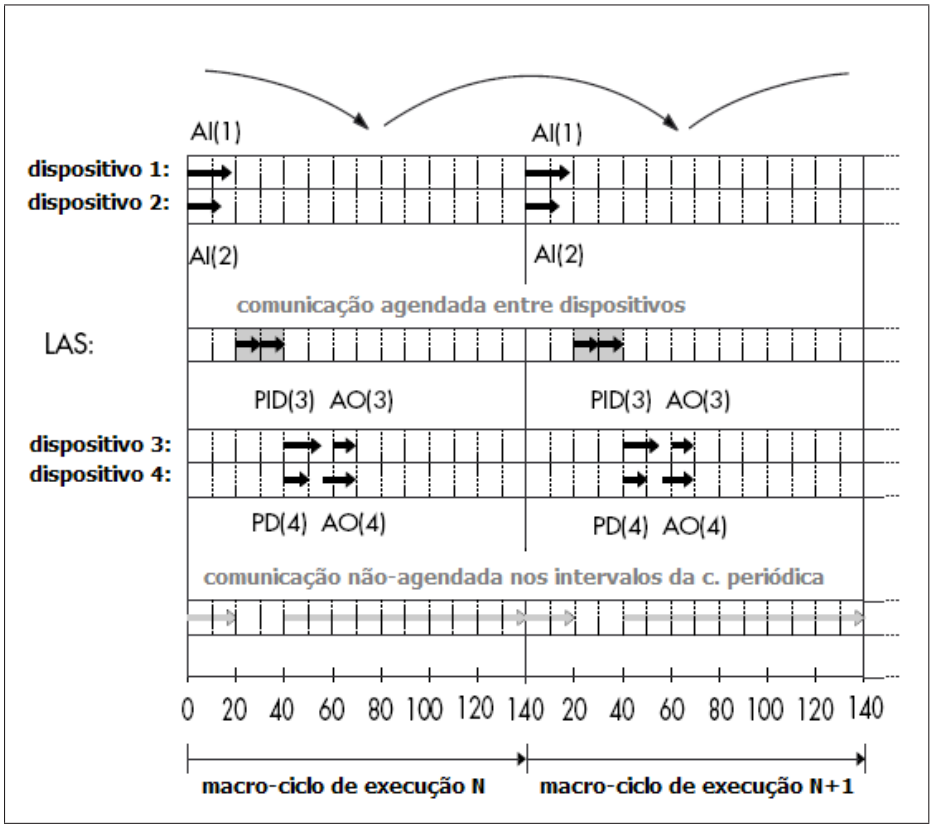

Figura 2.7: Ilustração do Escalonamento com Transmissões Síncronas e Assíncronas

É possível observar que o *loop* de controle acessa o barramento por um instante de tempo muito pequeno comparado ao macro-ciclo total de execução. Logo, faz-se possível a utilização do barramento de comunicação para outras atividades.

Transmissão de Dados Assíncrona

A parametrização de dispositivos e o diagnóstico de funcionamento dos mesmos são atividades que exigem transmissão de informação. Contudo, ao contrário das ações de controle de um processo, tais atividades não possuem característica de tempo crítico. Para

tais tarefas de comunicação, o *Foundation Fieldbus* provê uma opção para transmissão de dados assíncrona [Sam 2000].

As transmissões de dados assíncronas são realizadas exclusivamente nos intervalos entre as transmissões agendadas. O LAS atribui permissão a um dispositivo para usar o barramento em uma tarefa assíncrona, se não houver uma comunicação síncrona ativa.

A permissão para tal operação ocorre quando o LAS realiza passagem de um *token* a um dispositivo através de um comando PT (*Pass Token*). A entrega do *token* pode ser realizada para qualquer instrumento que esteja na *Live List* (lista com os dispositivos ativos no barramento - figura 2.8). A utilização do barramento se dá até o retorno do *token* para o LAS, ou o alcance de um tempo limite para a utilização do objeto.

A lista dos dispositivos é continuamente atualizada pelo LAS, através de comandos PN (*Probe Node*) para os endereços que ainda não fazem parte da lista. Se um dispositivo retorna um comando PR (*Probe Response*), ele é adicionado a lista, podendo receber o *token* para alguma tarefa assíncrona [Sam 2000].

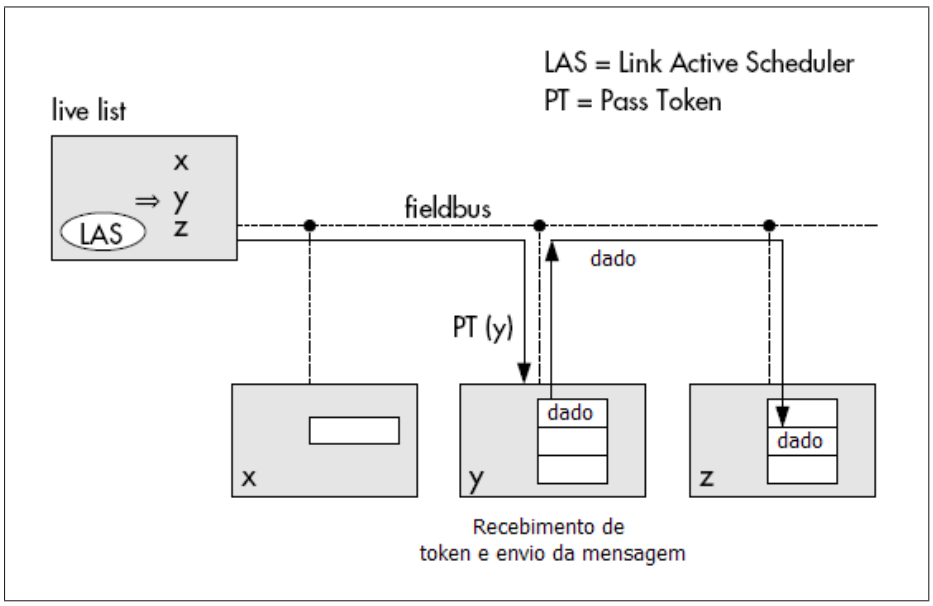

Figura 2.8: Transmissão de Dados Assíncrona com “Token”

A atualização da lista com os dispositivos ativos é enviada constantemente pelo barramento, visto que na necessidade de troca do mestre da rede, não haverá perda de informação.

Agendamento da Comunicação

O LAS segue fielmente o escalonamento (figura 2.9) para evitar que a comunicação assíncrona através de passagem de *token*, assim como comandos TD ou PN, não interfiram na comunicação síncrona.

Antes de cada operação, o LAS consulta a lista de transmissão para checar por alguma transmissão agendada. Se for o caso, ele espera (modo *idle*) por, precisamente, o tempo agendado e então envia o comando *Compel Data* (CD) para ativar a operação.

No caso de não haver comunicações agendadas e existir tempo disponível para operações extras, o LAS envia um dos comandos a seguir: O PN, para procurar por novos dispositivos; um comando TD para que todos os dispositivos estejam exatamente no mesmo tempo; ou o comando PT para passar o token da comunicação assíncrona. Esse ciclo se repete, começando sempre pela checagem na lista de transmissão [Sam 2000].

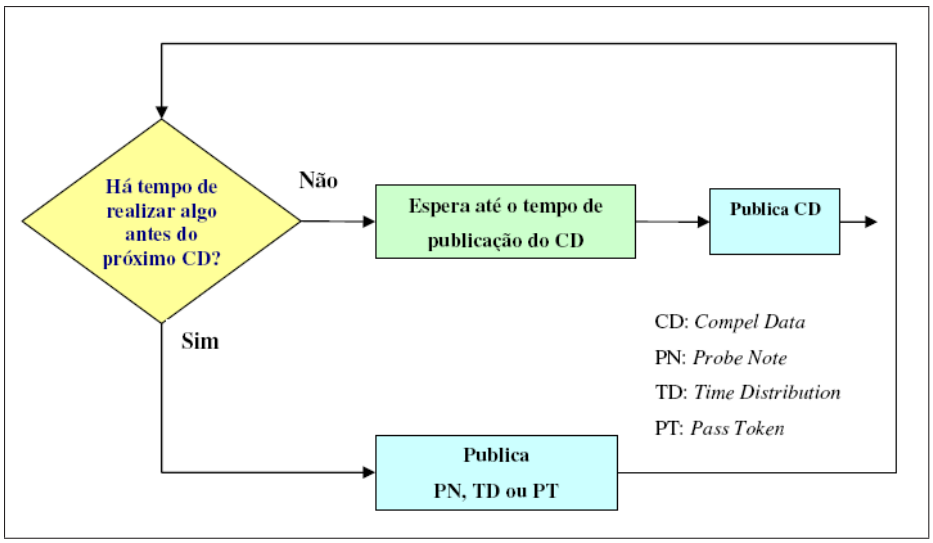

Figura 2.9: Fluxograma do Algoritmo do LAS

Camada de Aplicação (FAS + FMS)

A subcamada intermediária de acesso aos serviços de rede (FAS) e subcamada superior de montagem de mensagens (FMS) formam a *interface* entre a camada de enlace de dados da pilha de comunicação e a camada de aplicação do usuário. O serviços dessas subcamadas são transparentes para o usuário final, mas de extrema importância para a performance do sistema comunicação.

A FAS cria “Relações de Comunicações Virtuais” (VCR), as quais são utilizadas pela

subcamada superior. Tais relações consistem na descrição de diferentes tipos de processos de comunicação. Os três tipos de comunicação podem ser vistos na tabela 2.3.

Tabela 2.3: Tipos de Comunicação na FAS

Modelo Cliente/Servidor	*Report Distribution*	*Publisher/Subscriber*
Operações de Comunicação	Notificação de Eventos Envio de Alarmes	Publicação de Dados
Mudança de *Setpoint*	Envio de Alarme do Processo para Operador	Envio de Variável de um Transmissor para função PID
Tarefa Assíncrona	Tarefa Assíncrona	Tarefa Síncrona

A FMS provê serviços para padronização da comunicação. No caso, os tipos de dados são relacionados a modos de comunicação distintos. Nessa subcamada, há a definição dos Dispositivos de Campo Virtuais (VFD), os quais consistem em definições associadas a dispositivos, contendo a especificação de dados e objetos que são suportados pelo instrumento. Para tal, cada VFD está relacionado a descritores de objeto que são arquivos indexados com especificação de tipos de dados.

2.3.3 Camada de Aplicação do Usuário

O critério mais importante para aceitação no mercado da tecnologia *Fieldbus* é a interoperabilidade, permitindo que diversos instrumentos de fabricantes distintos possam ser conectados. Outra característica associada à primeira é a de intercambialidade, a qual permite a troca de dispositivos de diferentes fabricante sem o comprometimento da operação em questão.

Tais características exigem a especificação de um protocolo aberto que defina funções uniformes para dispositivos e *interfaces* de aplicações padronizadas. Para tal, o protocolo *Foundation Fieldbus* baseia-se em um modelo de blocos e no conceito de descritores de dispositivos [Berge 2001].

Modelo de Blocos

O *Foundation Fieldbus* associa todas as funções e dados dos dispositivos a três tipos básicos de blocos [Foundation n.d.]: Os blocos de recurso (*Resource Blocks*), blocos transdutores (*Transducers*) e os blocos funcionais (*Function Blocks*).

O bloco de recurso descreve características específicas do dispositivo *Fieldbus*, como o seu nome, fabricante, número serial e as versões de *hardware* e *firmware*.

Os blocos funcionais descrevem as funções que podem ser associadas a cada dispositivo e o modo de como serão acessadas. O escalonamento das tarefas de comunicação síncrona é baseado na organização desses blocos. Cada bloco realiza uma tarefa específica e, para isso, possui um conjunto de entrada e saída definido. Alguns blocos funcionais foram apresentados na tabela 2.1.

Os blocos transdutores expandem a complexidade e as aplicações possíveis de um dispositivo. Seus dados influenciam os parâmetros de entrada e saída dos blocos funcionais. Eles podem ser usados para calibrar, deslocar medidas, posicionar dados, linearizar características e converter unidades físicas.

Ao lado dos três tipos de blocos, os seguintes objetos também são definidos no modelo de blocos:

- *Link Objects* - definem ligações entre os blocos funcionais, sendo essas internas num dispositivo ou distribuídas na rede;
- *Alert Objects* - permitem reportar alarmes ou eventos na rede;
- *Trend Objects* - permitem o acompanhamento dos dados associados aos blocos funcionais para análise em níveis gerenciais da planta;
- *View Objects* - conjuntos de grupos de dados predefinidos que podem ser utilizados para visualização mais rápida de determinadas tendências.

A figura 2.10 exemplifica o modo como os blocos funcionais podem ser ligados através dos *links objects*. Na figura, são realizadas ligações entre um bloco de entrada analógica, um bloco de controle PID e um bloco de saída analógica. Tal configuração é capaz de realizar o controle de um determinado processo real.

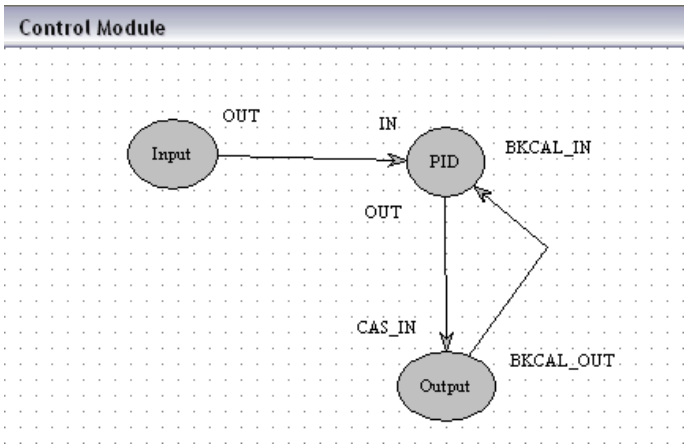

Figura 2.10: Exemplo de blocos funcionais e links lógicos numa estratégia de controle

No capítulo 4 serão apresentadas algumas características dos principais blocos funcionais usados na concepção do trabalho, assim como o modo pelo qual os *links* entre os blocos foram manipulados.

Descrição de Dispositivos

Durante a inicialização ou no gerenciamento, um sistema de comunicação aberto deve garantir que computadores de alto nível (supervisão) possam: acessar todos os dados dos dispositivos de campo e ter o próprio controle de como fazer isso.

As descrições de dispositivos (DD) contêm a informação necessária para preencher esses requisitos. Elas fornecem a informação necessária para entender o significado dos dados e apresentá-los ao operador [Fie 2003].

Para as funções básicas dos dispositivos, o *Foundation Fieldbus* usa diferentes blocos funcionais e blocos transdutores. Por isso, diferentes DD estão disponíveis e podem ser obtidos a partir da *Fieldbus Foundation*. Dispositivos FF podem interpretar e mostrar os dados e funções desses blocos padronizados, assim como apresentá-los ao usuário através de uma interface de operação.

Se um fabricante implementa funções e parâmetros extras em um dispositivo, ele deve definir o conteúdo, o acesso e a representação em uma descrição de dispositivo completa (estendida). Somente com o DD completo o dispositivo pode ser operado e aplicado por inteiro.

As descrições de dispositivos são escritas usando a *Device Description Language* - DDL (Linguagem de Descrição de Dispositivos) em forma de arquivo texto. Este arquivo é, então, convertido e distribuído via Internet. Se o fabricante registrou o seu dispositivo e seu respectivo DD à FF, então o DD pode ser obtido também, a partir da *Fieldbus Foundation* [Sam 2000].

2.4 OLE for Process Control - OPC

Apesar de não estar associado diretamente ao padrão digital FF, a tecnologia OPC será citada nesta seção, visto que através de um cliente OPC será realizada a reconfiguração das estratégias instanciadas nos dispositivos da rede industrial.

No caso, um dos dispositivos da rede didática utilizada para validação da proposta (será melhor detalhada no capítulo 4) servirá como servidor OPC, apresentando todas as variáveis disponíveis na rede industrial. Logo, através de um aplicativo que implementa um cliente OPC, faz-se possível a manipulação de tais variáveis alterando características

dos blocos funcionais e *links* lógicos existentes entre eles.

O padrão OPC estabelece as regras para que sejam desenvolvidos sistemas com *interfaces* padrões para comunicação dos dispositivos de campo (CLPs, sensores, atuadores) com sistemas de monitoração, supervisão e gerenciamento [Costa 2006].

Dentre as especificações do padrão desenvolvidas, alguns exemplos podem ser citados:

- OPC *Overview* - Descrição geral dos campos de aplicação das especificações OPC;
- OPC *Common Definitions and Interfaces* - Definição das funcionalidades básicas para as demais especificações;
- OPC *Data Access Specification* - Definição da *interface* para leitura e escrita de dados de tempo real;
- OPC *Alarms and Events Specification* - Definição da *interface* para monitoração de eventos.

Estas especificações têm a finalidade de orientar os desenvolvedores para a implementação das aplicações cliente e servidor.

O padrão baseia-se na tecnologia OLE (*Object Linking and Embedding*) desenvolvida pela Microsoft em meados de 1990, para suprir a necessidade de se integrar diferentes aplicações dentro da plataforma *Windows* [Costa 2006].

Dois grandes módulos são implementados pelo padrão: OPC *Server* e OPC *Client*. Enquanto o OPC *Server* especifica *interfaces* padrão de acesso direto aos equipamentos ou aplicações, o OPC *Client* especifica a *interface* padrão para as aplicações terem acesso aos dados coletados.

Para o caso específico, a DFI302 (LAS da rede didática que será apresentada no capítulo 4) é responsável pelo acesso aos demais dispositivos da rede e atua como servidor OPC para disponibilizar os dados coletados.

CAPÍTULO 3

Redes Neurais Artificiais

3.1 Introdução

O presente trabalho, como citado no capítulo 1, tem como objetivo a criação de uma metodologia para a realização de uma reconfiguração dinâmica das estratégias distribuídas em uma rede de dispositivos *Foundation Fieldbus*. Tais dispositivos, com a evolução tecnológica, deixaram de ser meros transdutores e passaram a realizar processamento interno mais complexo.

Como mostrado em [Costa 2006] [Lima 2004] [Marangoni 2005], foram realizadas diferentes configurações para instrumentos *Fieldbus* para atender diferentes aplicações, no caso, ações de controle distribuído e filtragem de sinais.

Para o trabalho em questão, a intenção é acrescentar a capacidade a uma rede industrial de realizar uma auto-configuração da estratégia distribuída sem a necessidade da interferência geral realizada no processo. Com isso, algumas aplicações como as realizadas em [Zanata 2005] [Fernandes 2007] podem ser incorporadas a um processo de maneira automática.

A ferramenta computacional base para tais aplicações nos processos da industria será a Rede Neural Artificial. As propriedades da ferramenta que a tornam compatível com as necessidades serão apresentadas abaixo. Já o respaldo para o uso das redes em um sistema *Fieldbus* pode ser observado em [Silva 2005].

Contudo, apesar da escolha da rede neural artificial como elemento de processamento nos dispositivos, não há a exclusão de outras ferramentas presentes no contexto dos sistemas inteligentes. Um dos exemplos utilizados para obtenção de resultados no capítulo 5 está associado a alteração de uma estratégia de escalonamento com lógica *fuzzy* e uma rede neural.

3.2 Definição e Propriedades das Redes Neurais

O desenvolvimento das redes neurais artificiais, desde o seu surgimento, vem sendo impulsionado pelo fato de que o cérebro humano processa informações de maneira diferente de um computador digital convencional. O cérebro é um computador (sistema de processamento de informação) extremamente complexo, não-linear e paralelo [Haykin 2004]. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar seus processamentos de maneira muito eficiente.

Na sua forma mais geral, uma rede neural artificial é um modelo matemático construído para tentar modelar a maneira como o cérebro realiza uma tarefa particular ou funções de interesse; essa rede é geralmente implementada a partir de componentes eletrônicos ou

simulada através da programação em um computador digital. Para alcançarem um bom desempenho, as redes neurais empregam uma interligação maciça de células computacionais simples, denominadas neurônios.

A seguinte definição pode, então, ser empregada para tal ferramenta [Haykin 2004]:

"Uma rede neural é um processador paralelamente distribuído constituído de unidades de processamento simples, que tem a capacidade natural de armazenar conhecimento experimental e torná-lo disponível para sua utilização."

A semelhança com o cérebro humano pode ser destacada a partir de dois aspectos. Em primeiro lugar, o conhecimento da rede é obtido a partir de um processo de aprendizagem. Além disso, os pesos sinápticos (variáveis ponderadoras que representam a conexão entre neurônios) são utilizados para armazenamento da informação.

Um algoritmo de aprendizagem é capaz de alterar os valores do pesos sinápticos, tornando-se a forma de realizar o processo de aprendizagem da rede.

Além da estrutura paralelamente distribuída da rede neural, sua capacidade de aprender e generalizar também deve ser destacada. A generalização se refere ao fato de a rede produzir saídas adequadas para entradas, as quais não estavam presentes durante o processo de aprendizagem.

Outras propriedades são enumeradas a seguir [Haykin 2004]:

1. Não-linearidade: Uma rede neural formada por neurônios não-lineares permite efetuar relacionamentos não-lineares entre entradas e saídas.
2. Mapeamento de entrada-saída: a aprendizagem supervisionada, ou aprendizagem com um "professor" envolve a modificação dos pesos sinápticos de uma rede neural pela aplicação de um conjunto de amostras de treinamento rotuladas ou exemplos da tarefa. Cada exemplo consiste de um sinal de entrada único e de uma resposta desejada correspondente. Apresenta-se para a rede um exemplo escolhido ao acaso do conjunto, e os pesos sinápticos (parâmetros livres) da rede são modificados para minimizar a diferença entre a resposta desejada e a resposta real da rede, produzida pelo sinal de entrada, de acordo com um critério estatístico apropriado. O treinamento da rede é repetido por muitos exemplos do conjunto até que a rede alcance um estado estável onde não haja mais modificações significativas nos pesos sinápticos. Os exemplos de treinamento previamente aplicados podem ser reaplicados durante a sessão de treinamento, mas em uma ordem diferente. Assim, a rede aprende dos exemplos ao construir um mapeamento entrada-saída para o problema considerado.
3. Adaptabilidade: as redes neurais têm uma capacidade inata de adaptar seus pe-

sos sinápticos a modificações do meio ambiente. Em particular, uma rede neural treinada para atuar em um ambiente específico pode ser facilmente retreinada para lidar com pequenas modificações nas condições de operação do ambiente. Além disso, quando está operando em um ambiente não estacionário, uma rede neural pode ser projetada para modificar seus pesos sinápticos em tempo real.

4. Resposta a evidências: no contexto da classificação de padrões, uma rede neural pode ser projetada para fornecer informação não somente sobre qual padrão particular selecionar, mas também sobre a confiança ou crença na decisão tomada. Esta informação pode ser utilizada para rejeitar padrões ambíguos, caso eles estejam presentes, e com isso melhorar o desempenho de classificação da rede.
5. Tolerância a falhas: uma rede neural, implementada na forma física (em *hardware*) é inerentemente tolerante a falhas, ou capaz de realizar computação robusta, no sentido de que seu desempenho se degrada suavemente sob condições de operação adversas. Se um neurônio ou suas conexões são danificados, por exemplo, a recuperação de um padrão armazenado é prejudicada em qualidade. Contudo, devido a natureza distribuída da informação armazenada na rede, o dano deve ser extenso para que a resposta global da rede seja degradada seriamente. O que deve ser observado nessas condições, é uma degradação suave do desempenho em vez de uma falha seriamente comprometedora.

3.3 Modelo de um Neurônio Computacional

Os neurônios biológicos e o sistema nervoso são a inspiração das redes neurais artificiais. Porém, as RNAs são bem diferentes das redes neurais biológicas e muitas vezes, as semelhanças são mínimas. Modelar o sistema nervoso é um trabalho que vem sendo desenvolvido há muito tempo. Os primeiros trabalhos que impulsionaram o interesse foram o de McCulloch e Pitts, o de Hebb, e o de Rosenblatt [Haykin 2004].

As RNAs são formadas por várias unidades de processamento conhecidas como neurônios. O neurônio artificial possui uma estrutura bem mais simples do que a de um neurônio biológico. A figura 3.1 ilustra um modelo não-linear de um neurônio computacional.

Um neurônio artificial é basicamente constituído por um conjunto de sinapses, um somador e uma função de ativação, geralmente não-linear. Além disso, os conjuntos de entradas e saídas são análogos, respectivamente, aos dendritos e aos axônios do neurônio biológico.

As entradas do neurônio artificial, antes de serem propagadas até a saída do mesmo, são ponderadas pelos pesos sinápticos. O somador tem o papel de somar esses sinais de

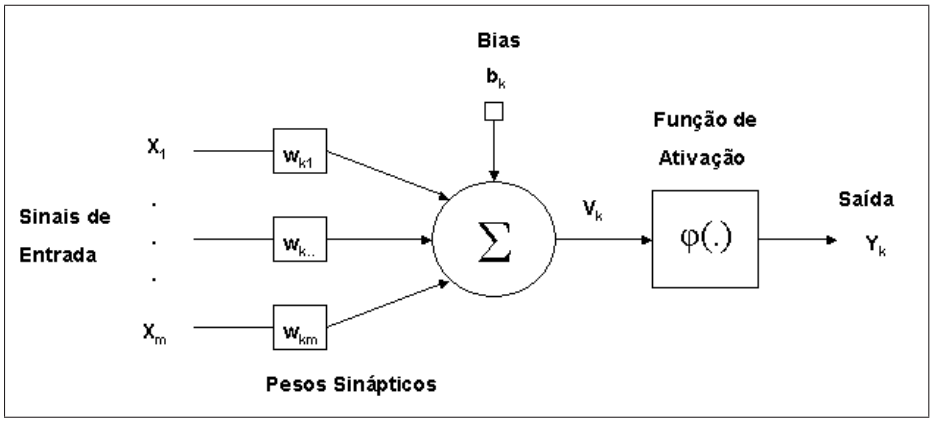

Figura 3.1: Modelo Não-linear de um Neurônio

entrada ponderados, tendo a função de um combinador linear. Já a função de ativação restringe a amplitude da saída de um neurônio e aplica a não linearidade.

No modelo da figura 3.1, existe um bias, b_k, que aumenta ou diminui a entrada líquida da função de ativação.

Uma modelagem matemática para um neurônio artificial k pode ser obtida através das equações 3.1 e 3.2,

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.1}$$

$$y_k = \phi(u_k + b_k) \tag{3.2}$$

em que x_1, x_2,..., x_m, são os sinais de entrada; w_{k1}, w_{k2},..., w_{km}, são os pesos sinápticos do neurônio k; u_k é a saída do combinador linear, devido aos sinais de entrada; b_k é o *bias*; $\phi(.)$ é a função de ativação e y_k é o sinal de saída do neurônio.

A função de ativação, $\phi(v)$, define a saída de um neurônio em relação ao campo local induzido v. As principais funções de ativação são: função de limiar, função linear por partes e função sigmóide. A escolha dessas funções depende da aplicação da rede neural. A figura 3.2 apresenta alguns exemplos de funções de ativação.

3.4 Arquitetura da Rede Neural

A maneira pela qual os neurônios de uma rede neural estão estruturados está intimamente ligada com o algoritmo de aprendizagem usado para treinar a rede. Para este trabalho não haverá um aprofundamento dos métodos de aprendizagem. Em relação às

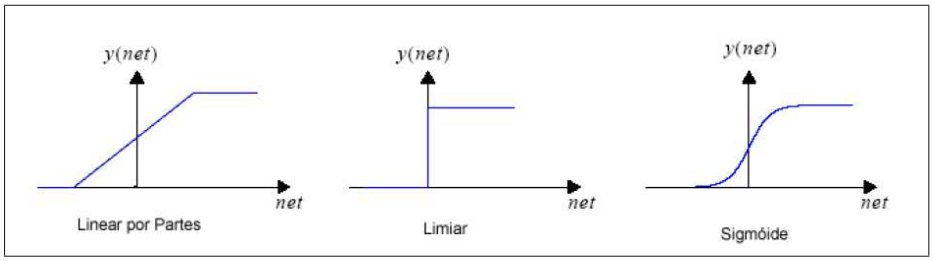

Figura 3.2: Funções de Ativação de um Neurônio

arquiteturas, pode haver uma divisão em três classes básicas: Redes Alimentadas Adiante com Camada Única, Redes Alimentadas Diretamente com Camadas Múltiplas e Redes Recorrentes.

A arquitetura que obteve êxito inicial, quando repassada para o padrão em blocos funcionais, foi a estrutura de Redes Alimentadas Diretamente com Camadas Múltiplas, também conhecida como *Perceptron* de Camadas Múltiplas.

3.4.1 Perceptron de Múltiplas Camadas

As redes *Perceptron* de Múltiplas Camadas (MLP) têm sido usadas com sucesso em diversas aplicações nas mais variadas áreas, como por exemplo, em reconhecimento de padrões, em processamento de sinais e em controle.

A estrutura de uma MLP consiste de nós em uma camada de entrada, de um conjunto de neurônios dispostos em uma ou mais camadas ocultas (ou intermediárias) e na camada de saída.

A figura 3.3 mostra a arquitetura de uma rede MLP com uma camada de entrada, duas camadas intermediárias e uma camada de saída.

Uma rede MLP é dita progressiva, ou *feedforward*, quando as saídas dos neurônio (em qualquer camada) se conectam apenas com os neurônios de camada sucessiva, ou seja, não possuem nenhum laço de realimentação. Dessa forma, o sinal de entrada é propagado camada a camada até chegar na camada de saída, ou seja, de forma progressiva.

A quantidade de nós na camada de entrada é determinada pela dimensão do espaço de observação. Já o número de neurônios da camada de saída é equivalente a quantidade associada ao objeto desejado.

Para o projeto de uma MLP, deve-se determinar o número de camadas ocultas, o número de neurônios em cada uma dessas camadas e definir os pesos sinápticos dos neurônios que constituem a rede.

Definir a quantidade de camadas ocultas e os neurônios que as compõem é uma tarefa

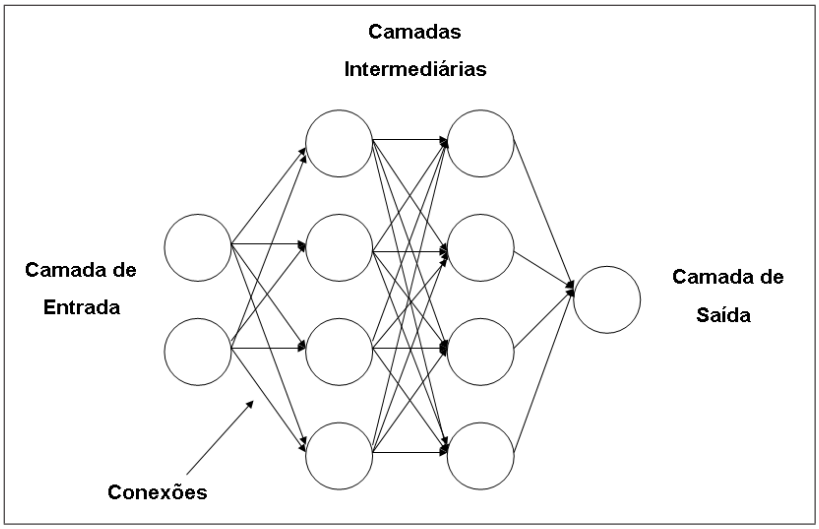

Figura 3.3: Exemplo de Arquitetura de uma Rede MLP

bastante complexa, pois não há regras determinadas para defini-los. Já a definição dos pesos sinápticos é feita utilizando algoritmos de treinamento supervisionado.

O algoritmo de treinamento mais conhecido na literatura é o Algoritmo da *Backpropagation*. O algoritmo *backpropagation* é baseado na minimização do erro usando métodos do tipo gradiente, em que o erro é retropropagado da camada de saída para as camadas intermediárias [Haykin 2004].

Esse algoritmo é constituído basicamente por dois passos: computação no sentido direto e computação no sentido reverso.

No primeiro passo do aprendizado (computação para frente), aplica-se um vetor de entrada aos nós de entrada da rede e seu efeito é propagado camada por camada até chegar à camada de saída, onde produz uma resposta ao vetor de entrada. Neste passo, os pesos sinápticos são mantidos constantes.

Na computação para trás, ajusta-se os pesos sinápticos através das regras de correção de erro. Basicamente, subtrai-se o padrão de saída desejado da resposta à excitação de entrada da MLP, que corresponde ao sinal de erro. Então, propaga-se esse sinal de erro através dos neurônios no sentido contrário ao que o vetor de entrada foi propagado no passo anterior, por isso o nome *backpropagation*. À medida que o erro é propagado, os pesos sinápticos são ajustados de forma que a resposta obtida pela MLP se aproxime ao máximo do padrão de resposta desejada [Haykin 2004].

No caso, da implementação em blocos funcionais padrões, todo o processo de aprendizagem (treinamento) é realizado de forma *offline*, ou seja, utiliza-se um *software*, o qual cria a estrutura da rede e aplica o algoritmo de aprendizagem. Por fim, a lista com pesos

é criada e deve ser enviada para os parâmetros correspondentes na rede *fieldbus*.

3.5 Implementação em Blocos Funcionais Padrões

Para a construção de uma rede neural artificial, são necessários vários neurônios artificiais interligados, formando a arquitetura desejada (a arquitetura implementada na referência [Silva 2005] é a *perceptron* de múltiplas camadas, mostrada na seção anterior).

Dentre os blocos funcionais (FF) padronizados, os mais importantes para a implementação da ferramenta em questão, são os blocos aritmético e caracterizador, que são diretamente utilizados no projeto de um neurônio artificial.

Através da configuração e interligação desses dois blocos funcionais (figura 3.4), pode-se obter um modelo de neurônio próximo ao mostrado na figura 3.1. As diferenças ficam por conta da função de ativação, visto que o bloco caracterizador só tem a capacidade de simular 20 pontos de uma determinada função [Silva 2005].

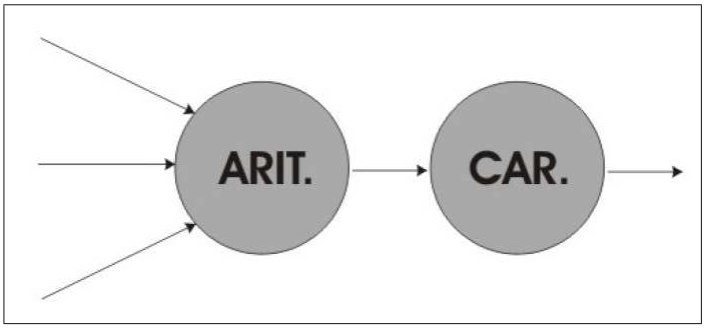

Figura 3.4: Neurônio Artificial Implementado no Ambiente Foundation Fieldbus

Como apresentado em [Silva 2005], os 20 pontos para simular as funções de ativação são obtidos através de um algoritmo genético que aproxima os vinte pontos da função desejada.

As demais características dos neurônios artificiais são simuladas a partir de parâmetros existentes nos blocos funcionais. Por exemplo, é possível configurar um bloco aritmético para receber, no máximo, três entradas ponderadas, realizar um somatório entre elas e, por fim, ajustar a saída através de um *bias*. Logo, a concepção da figura 3.1 está completa com base em blocos funcionais.

A interligação de neurônios para criação de uma rede MLP baseia-se na utilização dos *links* lógicos entre blocos funcionais. A figura 3.5 apresenta um exemplo de um modelo de rede conectada através dos *links* entre os blocos.

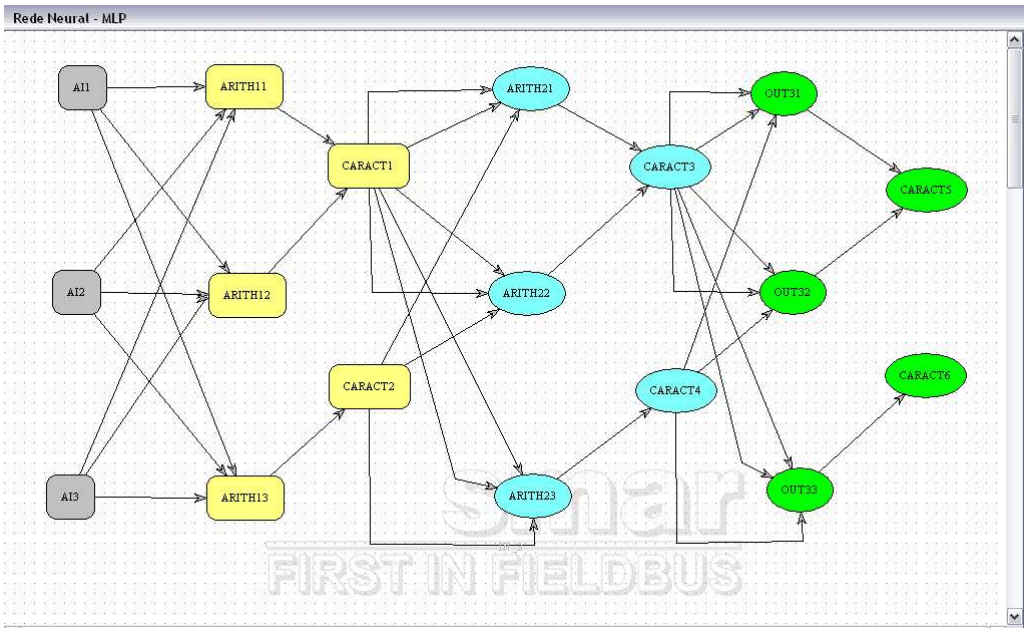

Figura 3.5: Rede Neural MLP implementada na rede FF

Tal modelo será usado posteriormente para a exemplificação da reconfiguração de estratégias baseadas em redes neurais, as quais possuam no máximo três neurônios em duas camadas ocultas. As diferentes cores expostas na figura sugerem a diferenciação das camadas existentes na MLP.

Os três primeiros nós pintados de cinza correspondem às três possíveis entradas da rede neural. A camada seguinte composta pelos nós pintados de amarelo corresponde à primeira camada oculta composta por três neurônios artificiais. Os nós “ARITH11”, “ARITH12” e “ARITH13” recebem três entradas cada, realizando a ponderação das mesmas a partir de parâmetros internos (pesos sinápticos). As saídas de cada nó podem ser ajustadas através de *bias* presente na saída do bloco funcional. Tais valores são repassados para os nós “CARACT1” e “CARACT2”, os quais simulam as funções de ativação baseados em 20 pontos da função real.

Cada bloco caracterizador só possuí duas entradas, portanto, a camada constituída por três neurônios requer a presença de dois caracterizadores. As duas camadas seguintes seguem a mesma lógica da explicada anteriormente. Os nós em azul implementam a segunda camada oculta e os nós em verde as saídas da rede.

No capítulo 4 será apresentada a estrutura interna de cada bloco funcional com suas variáveis e possíveis manipulações para ajuste dos *links* lógicos mostrados na figura 3.5.

CAPÍTULO 4

Proposta e Metodologia Experimental

4.1 Introdução

Como apresentado anteriormente, a proposta principal do trabalho consiste na criação de uma metodologia para alteração, em tempo de execução, de estratégias distribuídas em uma rede industrial *Foundation Fieldbus*.

A intenção inicial da proposta era alterar os *links* existentes entre blocos funcionais instanciados nas estratégias de controle da rede para que, em seguida, outros blocos que estivessem inutilizados nos dispositivos pudessem assumir suas tarefas (figura 4.1).

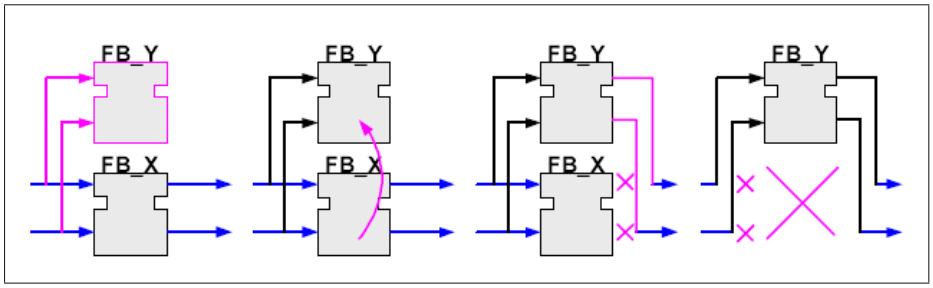

Figura 4.1: Exemplo de Reconfiguração dos Blocos Funcionais

Como visto na figura acima, o passo de reconfiguração é baseado em 4 fases, das quais duas apresentam nível crítico em relação ao desempenho temporal.

- Em primeiro lugar, deve-se realizar a parametrização do novo bloco funcional, assim como a conexão de suas entradas;
- A migração dos dados do bloco em execução é o próximo passo na tarefa de reconfiguração (nível crítico);
- Novos *links* com as saídas do novo bloco funcional devem ser criados (nível crítico);
- Por fim, o bloco inutilizado seria retirado da estratégia de controle [Strasser et al. 2006].

Uma das alternativas para realizar reconfigurações está associada a manipulação do macro-ciclo de execução da rede. No caso, durante a faixa de atuação de tarefas assíncronas, em que a rede não realiza ações de nível crítico em relação ao tempo de execução, pretendia-se instanciar os novos blocos funcionais para diferentes atividades distribuídas.

Contudo, para o estudo de caso proposto adiante, tal opção se tornou limitada diante de algumas restrições presentes no sistema de configuração da rede utilizada.

Como será apresentado no decorrer do texto, a base para a implementação do sistema de reconfiguração é uma rede industrial FF com dispositivos do fabricante brasileiro

SMAR. O *software* que realiza a configuração dos instrumentos da rede é o aplicativo *Syscon* (do mesmo fabricante dos dispositivos). Este programa, por consequência, gera também o algoritmo de escalonamento da rede em questão.

Em [Zerbetto 2007], é realizada uma análise do desempenho geral de sistemas de controle quando os mesmos são gerenciados por uma arquitetura de rede industrial *Foundation Fieldbus*. Neste trabalho, devido a necessidade de se encontrar um tipo de escalonamento ótimo para evitar influência no sistema de controle, observa-se que a ferramenta de configuração *Syscon* não permite a intervenção direta na política de escalonamento a ser utilizada no sistema de controle em rede, contudo, o macro-ciclo pode ser alterado (de forma limitada) através do dimensionamento do parâmetro *background traffic*, o qual corresponde ao tempo de execução das tarefas assíncronas.

Logo, torna-se inviável a tentativa de interferir no macro-ciclo de execução da rede para incorporar outras atividades que não estejam estruturadas através do algoritmo gerado pelo *software* de configuração.

[Zerbetto 2007] ainda ressalta algumas fórmulas apresentadas pelo fabricante SMAR para o caso de sistemas redundantes e não-redundantes, baseado na quantidade de blocos presentes em cada dispositivo usado no barramento, assim como nas ligações externas presentes na configuração de controle (vale lembrar que as ligações externas correspondem às ligações lógicas entre dispositivos diferentes, as quais necessitam do uso do barramento para transmitirem mensagens vinculadas ao sistema de controle definido na aplicação).

A estimativa do cálculo de macro-ciclo fornecido pelo *Syscon* consiste, portanto, nas três equações descritas abaixo [Smar n.d.]:

- *Background Traffic*: tempo usado para supervisão e mensagens assíncronas;

$$BT = (NumerodeDispositivos * NumerodeBlocos) * 30ms \tag{4.1}$$

- *Foreground Traffic*: tempo utilizado para *links* e controle;

$$FT = NumerodeLinks * 30ms \tag{4.2}$$

- *Macro-ciclo de Execução*: estimativa do macro-ciclo de execução consiste na soma dos tempos anteriores com uma margem de segurança de 20%.

$$MC = (BT + FT) * 1,2 \tag{4.3}$$

Em [Cicillini 2007], é realizado o desenvolvimento de um algoritmo para otimizar o escalonamento das mensagens de comunicação cíclica ou periódica. Tal implementação ocorre a partir de informações geradas pelo *Syscon*, o qual (em suas últimas versões) permite exportar as configurações dos dispositivos em um arquivo, cujo formato de extensão é XML.

No entanto, o algoritmo limita-se ao agendamento de apenas uma parcela do macro-ciclo, além de não haver uma forma de integrar tal otimização ao escalonamento gerado pelo *Syscon*. No caso, o padrão de comparação realizado em [Cicillini 2007] é baseado nas fórmulas citadas anteriormente.

Outra característica importante destacada em [Zerbetto 2007] ocorre devido à existência de diversos blocos funcionais e sua disponibilidade para a maioria dos transmissores. No caso, faz-se possível a implementação de uma mesma estratégia de controle de diferentes formas, considerando os mesmos blocos funcionais alocados de diversas maneiras.

A figura 4.2 apresenta um exemplo prático de estratégia de controle sendo realizada de duas formas distintas. O transmissor de pressão (LD302) envia um sinal de entrada analógico que é recebido através de um bloco funcional PID e, em seguida, após a computação do algoritmo de controle, um sinal de saída analógico é enviado através do posicionador de válvula (FY302).

A partir do *Syscon*, é possível configurar a estratégia de duas formas, já que os dois dispositivos apresentam um bloco funcional PID. Logo, como pode ser visto na figura, o macro-ciclo de execução pode ter configurações distintas para um mesmo algoritmo de controle, pois há a possibilidade de variação no número de *links* externos entre os blocos.

Outra restrição encontrada na proposta inicial do trabalho consiste no estado em que dispositivos da rede entram para realizar um *download* de configuração, visto que não há um modo de manter os dispositivos em operação contínua quando existe a intenção de reconfigurá-los a partir de uma nova estratégia, a qual foi desenvolvida no *Syscon*.

Só é possível ajustar alguns parâmetros que permitem escrita contínua durante o tempo de execução. A inserção de novos *links* lógicos exige uma parada no modo de operação dos instrumentos.

4.2 Proposta

Diante da impossibilidade de realizar uma realocação de uma estratégia com blocos funcionais diferentes daqueles que foram pré-instanciados, assim como alterar o macro-ciclo de execução, adotou-se uma alternativa diferente.

A partir de uma estratégia fixa na rede, torna-se possível alcançar diferentes aplicações

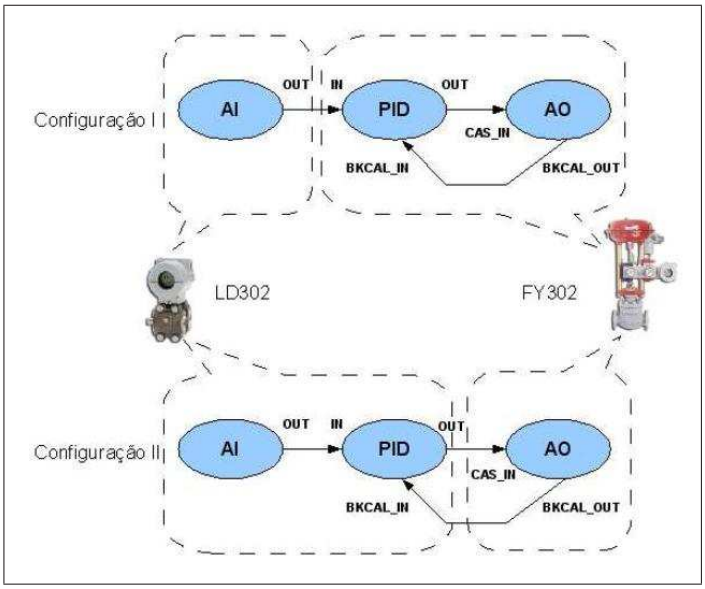

Figura 4.2: Exemplo de alocações de blocos funcionais para uma estratégia de controle

com base em alguns artifícios usados nos blocos funcionais padrões.

Como exemplo, que será demonstrado no capítulo 5, uma rede neural foi instanciada na rede industrial com uma arquitetura máxima 3-3-3-3 (3 entradas, 3 neurônios nas duas camadas intermediárias e 3 saídas - figura 3.5). Através de um cliente OPC, é possível alterar a arquitetura da rede, ajustando pesos sinápticos e modificando as funções de ativação.

Em [Prayati et al. 2004], é apresentada uma metodologia (figura 4.3) para o desenvolvimento de aplicações em tempo real de controle distribuídas com foco na alocação de tarefas para o sistema em questão.

Tal metodologia é baseada em 7 passos que vão desde a especificação da aplicação até a configuração do sistema.

Fazendo um paralelo com a configuração da rede industrial *Foundation Fieldbus* para atender mais de uma funcionalidade, são descritos abaixo os 7 passos de acordo com a proposta citada anteriormente, em que a partir de uma pré-configuração base é possível atender mais de uma aplicação.

1. Especificação da Aplicação: as necessidades impostas pelo processo industrial em questão devem servir para a concepção das diferentes aplicações. No caso específico dos dispositivos de campo FF, pode-se observar um bom desempenho para aplicações cujas as ferramentas computacionais foram redes neurais artificiais distribuídas através da rede industrial, por exemplo;

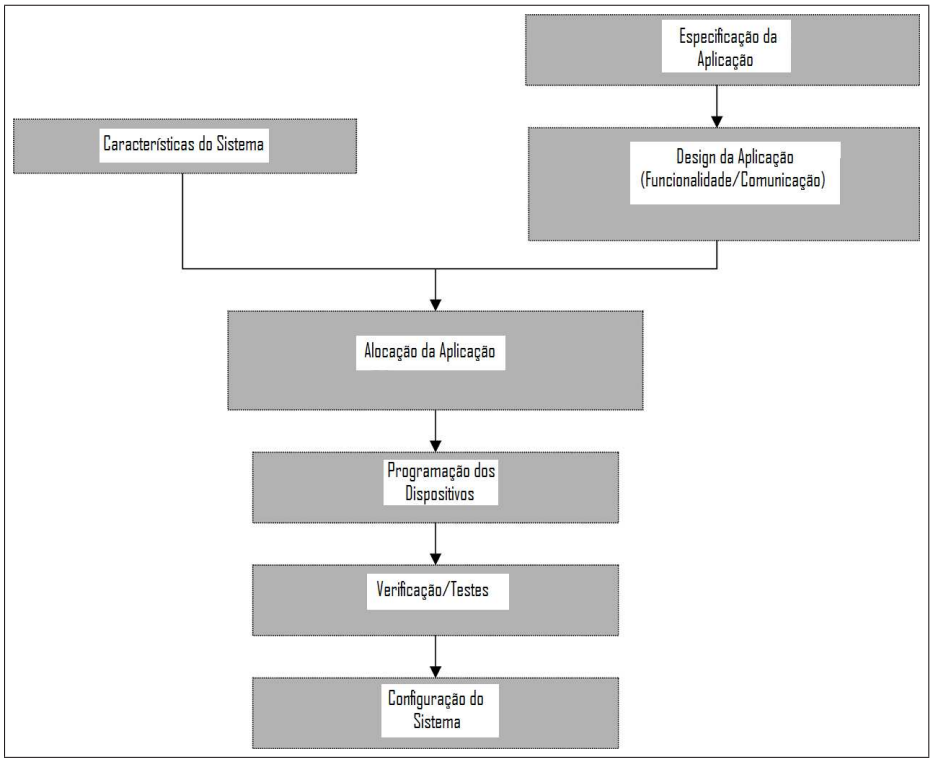

Figura 4.3: Metodologia para Desenvolvimento de uma Aplicação Distribuída

2. Design da Aplicação: em termos funcionais, o processo para criação da aplicação está diretamente ligado à arquitetura da rede escolhida para sua implementação em termos de blocos funcionais. No exemplo citado de uma reconfiguração a partir de uma arquitetura de rede neural, deve-se especificar uma arquitetura de rede mínima para atender às diferentes funcionalidades desejadas. Assim como, se há a intenção de alterar uma estratégia de controle de um controle convencional para um esquema com lógica fuzzy, deve-se prever todas as necessidades para que as duas funcionalidades sejam atendidas;
3. Características do Sistema: uma análise geral da rede industrial é essencial para o passo seguinte de alocação da aplicação nos dispositivos presentes na rede. No caso, a quantidade de dispositivos e a capacidade corrente de cada um deve ser levada em consideração;
4. Alocação da Aplicação: tal passo está diretamente ligado ao anterior. Após realizar uma análise detalhada dos dispositivos associados ao barramento, deve-se fazer a ligação entre as características da rede neural (neurônios, camadas e pesos) com os blocos dos instrumentos disponíveis, por exemplo;
5. Programação dos Dispositivos: os três últimos passos estão ligados às caracterís-

ticas de funcionamento do protocolo *Foundation Fieldbus*. Após a alocação dos componentes da aplicação com os dispositivos específicos, deve se programar cada um deles com as informações necessárias;

6. Verificação/Testes: esta fase é um item de segurança dentre os ítens para configuração da rede industrial. No caso, realizar uma revisão nos blocos dos dispositivos e nas ligações entre eles para verificar consistência da aplicação;
7. Configuração do Sistema: com os outros passos realizados com sucesso, o sistema pode iniciar sua operação com a aplicação configurada.

Os dois primeiros passos ressaltados na metodologia acima serão destacados no decorrer do desenvolvimento do trabalho.

Um dos cenários para validação da proposta consiste em uma aplicação de controle do processo especificado no anexo A. As equações que regem a dinâmica do sistema sugerido trazem a necessidade de algum tipo de escalonador para ponderar os ganhos de dois controladores PID implementados através de blocos funcionais.

São especificadas duas aplicações distintas, as quais realizam o escalonamento sugerido de diferentes formas. A concepção de tais funcionalidades através dos blocos configurados na rede industrial segue os 2 primeiros passos indicados acima.

As duas aplicações são pré-instanciadas na rede e, através da manipulação dos *links* existentes na estratégia de controle, são realizadas as reconfigurações entre os dois tipos de escalonadores.

4.3 Metodologia Experimental

Nesta seção será detalha toda a metodologia experimental para a realização de testes da proposta em questão. Destaca-se a utilização de uma rede industrial didática *Foundation Fieldbus* utilizada em todo o desenvolvimento do trabalho.

4.3.1 Ambiente Híbrido

Para a validação do trabalho, havia a necessidade da utilização de uma planta de um processo associada a um rede industrial. Como contribuição destaca-se, no caso, a criação de um ambiente híbrido, contendo a simulação de um processo físico e uma rede industrial didática presente nas instalações do LAMP.

A idéia para criação do ambiente surgiu da necessidade de se testar os sensores de *software* criados pelo grupo de pesquisa do laboratório [Costa et al. 2008] [Rodrigues

et al. 2008]. No caso, devido à inexistência de um processo real nas instalações da Universidade, formulou-se uma arquitetura que contivesse um *software* para simulação do processo e o *hardware* necessário para que o processo estivesse integrado aos dispositivos da rede.

4.3.2 Arquitetura do Ambiente

O ambiente proposto é composto por três partes distintas: uma rede industrial didática FF formada por 8 dispositivos e um gerenciador do barramento, uma simulação desenvolvida em um aplicativo específico e as placas de aquisição e circuitos conversores necessários para integração entre os dois primeiros ítens.

A figura 4.4 apresenta a arquitetura citada anteriormente.

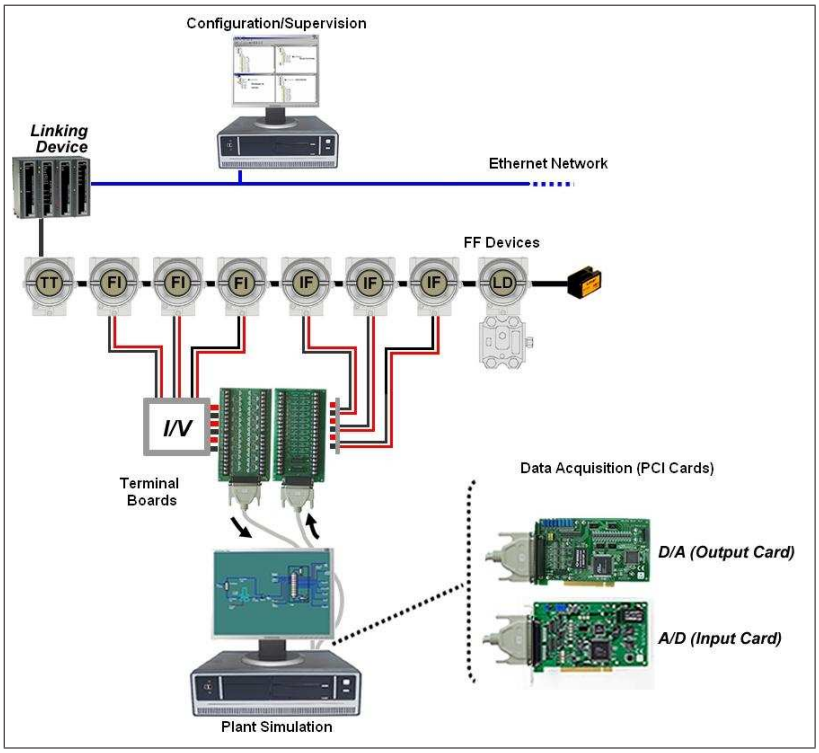

Figura 4.4: Arquitetura do Ambiente Híbrido

4.3.3 Rede Industrial Didática

A rede industrial é formada por dispositivos da SMAR. Há a presença de um sensor de pressão (LD302) e um transmissor de temperatura (TT302), instrumentos que fazem conversão do padrão digital para *loop* de corrente, assim como o processo contrário (FI302

e IF302, respectivamente) e um dispositivo que serve como *bridge* entre o barramento de campo e a rede *Ethernet* (DFI302 - *Linking Device*). No caso, o sistema pode ser configurado através do *software Syscon*, o qual faz parte do *System302* (sistema proprietário da SMAR para configuração e testes numa rede *Fieldbus*) [Smar n.d.].

Os conversores FI302 e IF302 assumem os papéis dos sensores na prática, visto que não há a presença das variáveis físicas (temperatura, pressão, etc.) com a utilização da simulação. Logo, tais variáveis passam a ser simuladas por sinais analógicos de corrente e tensão.

As características de funcionamento interno dos conversores de padrão serão detalhadas abaixo, assim como alguns detalhes da *DFI302*.

Ponte Universal Fieldbus - DFI302

A DFI302 é dispositivo multifunção que incorpora componentes de *hardware* e *software* para gerenciar, monitorar, controlar, manter e operar uma planta industrial. A DFI302 executa a maioria das funções exigidas pelo sistema de controle, resultando em um número reduzido de componentes adicionais [Smar n.d.].

O equipamento adquirido pelo laboratório de pesquisa possui 4 módulos principais: DF50 (módulo fonte), DF51 (processador), DF52 (módulo fonte para a rede FF) e o DF53 (impedância de linha). O dispositivo pode ser visualizado na figura 4.5.

Figura 4.5: Ponte Universal Fieldbus - DFI302

A configuração da DFI é facilitada pois utiliza a mesma idéia de blocos funcionais dos instrumentos de campo (blocos funcionais do padrão *Foundation Fieldbus*), desta forma, utilizando-se o *software* de configuração dos blocos funcionais dos instrumentos, pode-se configurar também as funcionalidades da DFI.

Outras características importantes são ressaltadas abaixo [Smar n.d.]:

- Interoperável com instrumentos e softwares de diferentes fabricantes devido a utilização de padrões abertos como FF e OPC;

- Conecta-se a equipamentos já existentes através de E/S convencionais e comunicação *Modbus* via RS232 (padrão de comunicação serial) ou *Ethernet*.
- Redundância em vários níveis (Servidor OPC, LAS, *Links* H1);

Conversor Loop de Corrente para Padrão FF - IF302

O IF302 é um conversor destinado a interligar transmissores analógicos com uma rede *Fieldbus*. O IF302 recebe até três sinais de corrente tipicamente de 4 - 20mA ou 0 - 20mA e torna-os disponíveis para um sistema *Fieldbus* [Smar n.d.].

A figura 4.6 apresenta o modo de conexão com os transmissores de corrente, além de sua configuração interna baseada em blocos.

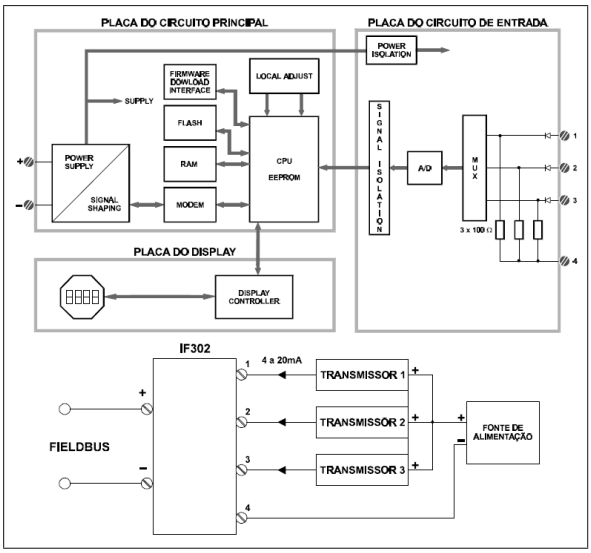

Figura 4.6: Diagrama de blocos do IF e o modo de ligação das entradas

A principal vantagem do dispositivo é poder servir como sistema de integração de equipamentos antigos (que ainda trabalham com *loop* de corrente) com os atuais equipamentos digitais, convertendo as informações analógicas em informações digitais (levando em consideração a unidade do dado transformado).

A configuração do equipamento é realizada através da parametrização de seus blocos funcionais, ou no próprio equipamento, através de uma chave magnética.

Conversor Padrão FF para Loop de Corrente - FI302

O FI302 segue a mesma lógica do dispositivo anterior. É um conversor destinado a conectar sistemas *Foundation Fieldbus* com atuadores e posicionadores de válvulas de

controle 4 - 20mA. O FI302 produz uma saída de 4 - 20 mA proporcional à entrada recebida pela rede FF [Smar n.d.].

Apesar da mesma quantidade de conexões que o IF302, o FI302 possui uma forma de conexão um pouco distinta. Há a necessidade de uma fonte externa para gerar a corrente e o equipamento apenas regula sua intensidade, baseado na variação do valor que recebe como entrada e na conversão que ele faz para a escala do instrumento que deseja controlar.

A figura 4.7 apresenta o modo de conexão de suas saídas, além de sua configuração interna baseada em blocos.

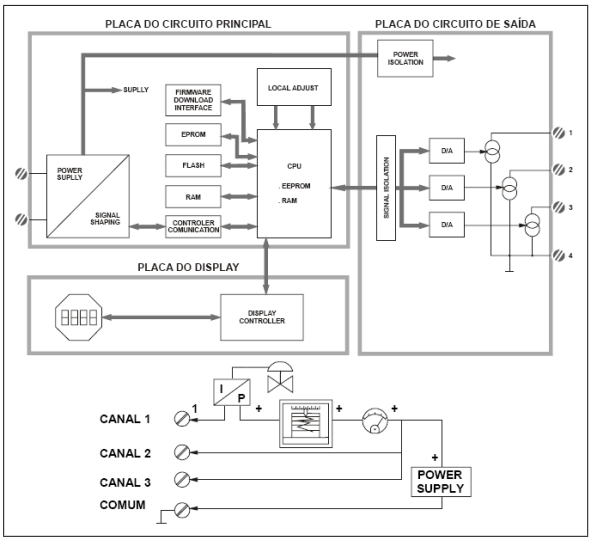

Figura 4.7: Diagrama de blocos do FI e o modo de ligação das saídas

O modo de configuração do FI302 é idêntico ao do IF302 (através de *software* ou de chave magnética usada no próprio instrumento).

4.3.4 Hardware de Integração

O *hardware* de *interface* corresponde às placas de aquisição de dados (uma placa conversora digital-analógico e outra conversora de analógico-digital) e a placa desenvolvida no laboratório para conversão de *loop* de corrente para tensão.

A figura 4.8 apresenta as *interfaces* de *hardware* e em seguida é realizada uma descrição de cada componente.

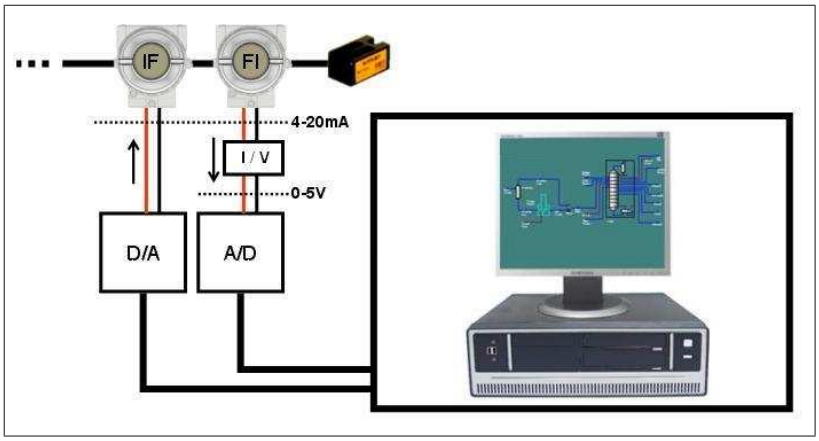

Figura 4.8: Interfaces de Hardware para Interconexão entre o Processo e a Rede Industrial

Placa Conversora Digital-Analógico (D/A)

A placa conversora D/A é do modelo PCI-1720U para barramento PCI de computador e provê 4 saídas D/A de 12 *bits* [Advantech n.d.].

Suas principais características são a capacidade de trabalhar com padrão unipolar (de 0-5V ou 0-10V) ou bipolar ($\pm$5V ou $\pm$10 V). Também pode trabalhar diretamente com *loop* de corrente (de 0-20mA ou de 4-20mA, o qual é o padrão utilizado nas indústrias e o mais importante para o trabalho).

A placa de aquisição pode ser observada na figura 4.9.

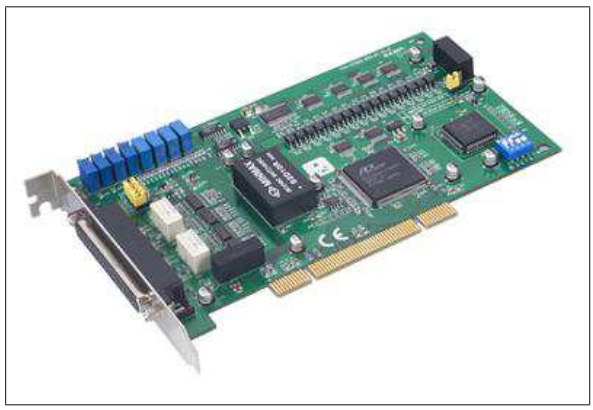

Figura 4.9: Placa D/A PCI-1720U

Placa Conversora Analógico-Digital (A/D)

A placa A/D é do modelo PCI-1713 também para barramento PCI e provê 32 entradas analógicas de alta velocidade de amostragem (100 kS/s ou seja 100000 amostras por segundo), utilizando uma resolução de 12 *bits* [Advantech n.d.] (figura 4.10).

Figura 4.10: Placa A/D PCI-1713

Por trabalhar apenas com padrão de entrada de tensão, faz com que seja necessário uma *interface* entre o FI302 e este dispositivo, visto que o dispositivo FF gera um *loop* de corrente como sinal de saída.

Placa Conversora de Loop de Corrente em Tensão

Esta placa foi desenvolvida com o objetivo de converter o loop de corrente de 4-20mA para um sinal de tensão de 0-5V. A placa possui seis circuitos de conversão, ou seja, pode converter até seis sinais ao mesmo tempo, utilizando uma alimentação de 12 + 12, ou seja, -12, terra e +12 V. Ela utiliza como componente principal o receptor RCV-420 da Burr-Brown [Lima 2004].

A figura 4.11 mostra a configuração do componente utilizado na placa de conversão.

Através dos três dispositivos de *hardware* comentados nas últimas seções, faz-se possível a comunicação da rede didática FF com uma estação que tenha a simulação ativa. Para isso, basta que a estação tenha acesso aos dados da placa de aquisição.

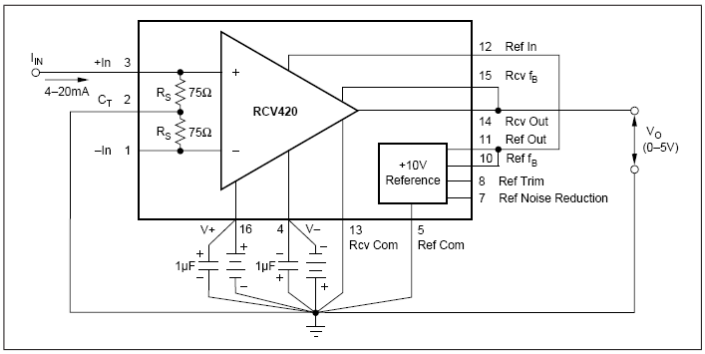

Figura 4.11: Circuito do RCV-420 configurado para conversão de loop de corrente em tensão

4.3.5 Software de Simulação

Em [Costa et al. 2008] e [Rodrigues et al. 2008], o ambiente híbrido para a concepção dos sensores de *software* é apresentado e o aplicativo base para a simulação dos processos da indústria petroquímica é o *ASPEN HYSYS*.

O programa oferece a simulação em regime permanente de modelos dinâmicos de plantas, permitindo a monitoração em tempo real do desempenho da planta, ocorrências de falhas no processo, atuação de controladores e qualidade de compostos.

Contudo, a utilização desta ferramenta não é essencial, pois o foco do trabalho não está direcionado para a complexidade dos processos envolvidos. Para a validação da proposta, o aplicativo escolhido para a simulação dos processos industriais foi o *Labview* da *National Instruments* [Labview n.d.].

Linguagem G

O *Labview* utiliza uma linguagem própria para o desenvolvimento de suas aplicações. A linguagem G é uma linguagem gráfica de programação de alto nível baseada no fluxo de dados através de um diagrama de blocos. Deste modo, ela dispensa o uso de qualquer formalismo sintático, típico das linguagens de programação convencionais, para a construção do código-fonte, facilitando bastante o trabalho do programador. De fato, não é necessário um conhecimento aprofundado em programação para se construir programas.

A representação gráfica do algoritmo e a depuração interativa oferecidas, permitem que a construção de programas seja feita de forma fácil e simplificada. Seguindo uma tendência cada vez mais presente na programação, a linguagem G segue o conceito de programação modular [Labview n.d.]. A figura 4.12 apresenta o trecho de código na

linguagem G para leitura/escrita de variáveis através do padrão OPC.

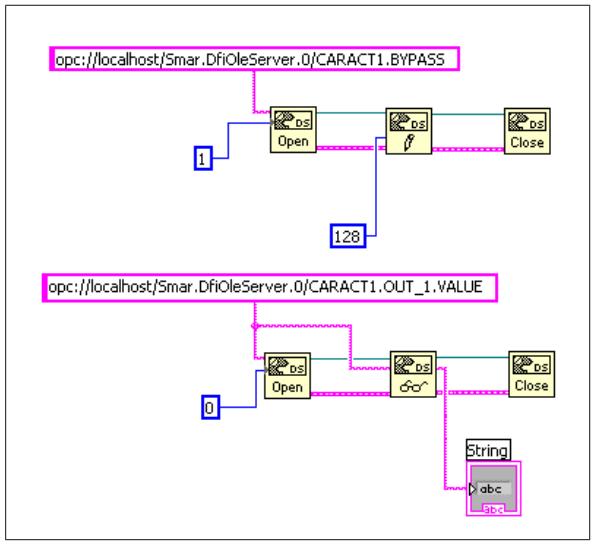

Figura 4.12: Trecho em Labview para leitura/escrita através do padrão OPC

Nas próximas seções serão demonstradas algumas aplicações desenvolvidas no *Labview*.

4.4 Exemplo de Estudo de Caso

Nesta seção será apresentado um exemplo de estudo de caso baseado na alteração de diferentes arquiteturas de rede neural a partir de uma estratégia pré-instanciada na rede didática FF.

Para tal, faz-se necessário entender como a rede neural base é pré-instanciada na rede industrial, com a descrição dos principais blocos funcionais utilizados, assim como, quais artifícios utilizados para realizar as reconfigurações.

Como comentado anteriormente, a ferramenta para configuração das estratégias de controle na rede *Fieldbus*, utilizada na implementação e testes, é o *software Syscon* da SMAR. Através desse programa, o qual se comunica com a DFI302, é possível enxergar todos os equipamentos presentes no barramento H1 e, usando a parametrização dos blocos funcionais, configurá-los.

Outra opção da ferramenta é a possibilidade de criar *links* entre os blocos funcionais utilizados, gerando dessa forma as estratégias distribuídas entre os dispositivos. As figuras 4.13 e 4.14 apresentam a implementação de duas redes neurais baseadas em blocos funcionais no *Syscon* correspondentes as funções seno e exponencial, respectivamente.

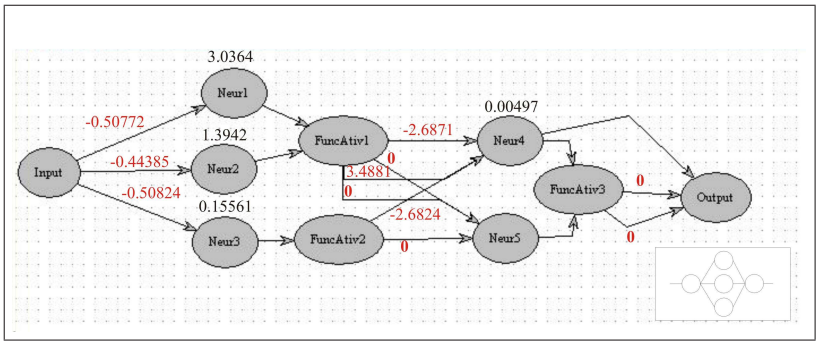

Figura 4.13: Rede Neural 1-3-1 simulando a função seno

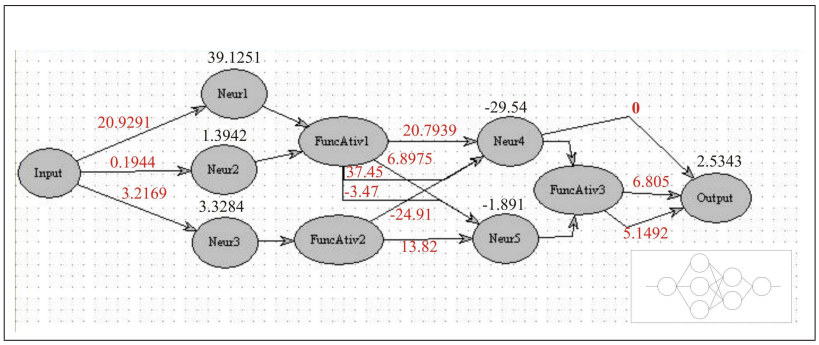

Figura 4.14: Rede Neural 1-3-2-1 simulando a função exponencial

Tais implementações são realizadas a partir de uma mesma arquitetura, levando em consideração os blocos funcionais pré-instanciados. Contudo, pode-se observar que, a partir da manipulação dos *links* lógicos e parâmetros internos dos blocos, é possível obter duas arquiteturas distintas.

A figura 4.13, por exemplo, corresponde a uma rede neural do tipo 1-3-1 (uma entrada, três neurônios na camada intermediária e uma saída). Já a figura 4.14 corresponde a uma rede neural cuja arquitetura é 1-3-2-1 (uma entrada, três neurônios na primeira camada intermediária, dois neurônios na segunda e uma saída).

O capítulo 3 apresenta, em sua última seção, o meio de utilização dos blocos funcionais aritmético e caracterizador para gerar a estrutura de um neurônio artificial. As características de funcionamento desses blocos, assim como outros utilizados nas implementações serão demonstradas a seguir.

4.4.1 Principais Blocos Funcionais Utilizados

Levando-se em consideração um estudo de caso baseado na reconfiguração de arquiteturas de redes neurais distribuídas nos instrumentos FF, faz-se necessário uma pequena explanação sobre as principais características do blocos funcionais utilizados. Para tal, serão apresentadas algumas informações sobre os blocos de entrada analógica, o bloco aritmético, o caracterizador de sinais e o de saída analógica.

Apesar de não fazer parte da estrutura apresentada na figura 3.5, o bloco funcional de controle PID também será levado em consideração, visto que é aplicável para situações em que a funcionalidade desejada é o controle do processo em questão.

Bloco de Entrada Analógica (Analog Input)

Este bloco obtém dados de um bloco transdutor através da escolha do canal e os disponibiliza em sua saída. Geralmente está associado a um instrumento de entrada de dados como o IF ou um sensor de pressão [Fie 2001]. O esquemático do bloco de entrada analógica pode ser visto na figura 4.15 [Smar n.d.].

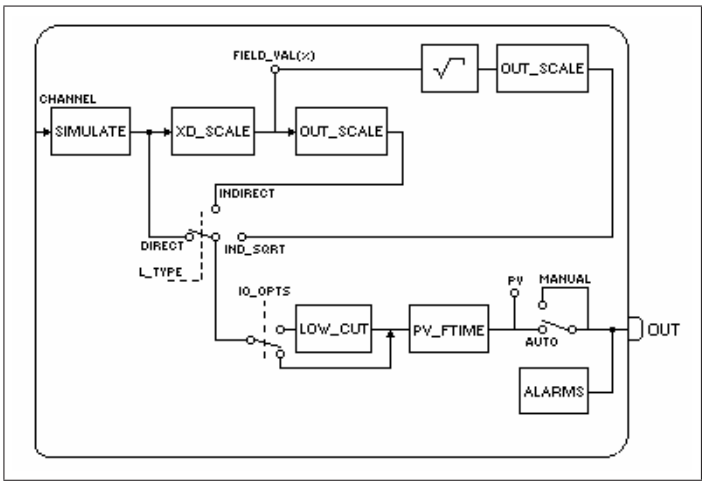

Figura 4.15: Esquemático do Bloco de Entrada Analógica

Os principais parâmetros ajustados são a variável *CHANNEL* que identifica o canal do bloco transdutor no qual ele ira realizar a leitura (o bloco transdutor é o que realiza a leitura direta das entradas dos instrumentos, estando presente em todos os instrumentos de campo), o *XD_SCALE* que é a escala da variável de entrada do instrumento, *OUT_SCALE* que determina a escala de saída do instrumento e *L_TYPE*, o qual determina a opção para converter ou não os dados do canal de entrada para a escala de sua saída.

Para o caso da rede neural da figura 3.5, os blocos de entrada analógica funcionam como as entradas da rede, sendo identificados como a primeira camada de nós.

Bloco Aritmético (Arithmetic)

O bloco aritmético foi desenvolvido para realizar cálculos sobre sinais provenientes dos sensores. Ele possui 5 entradas, sendo as duas primeiras utilizadas na sua função de extensão de *range* resultando em uma PV (variável do processo) e as outras três são utilizadas em combinação com a PV em funções matemáticas disponíveis no bloco [Fie 2001]. Cada uma das outras três entradas possui um ganho e um *BIAS* associado, isto para poderem efetuar correções nos valores recebidos dos sensores, enquanto que a saída também possui um ganho e um *BIAS* para ajustes posteriores dos cálculos, como pode ser visto na figura 4.16 [Smar n.d.].

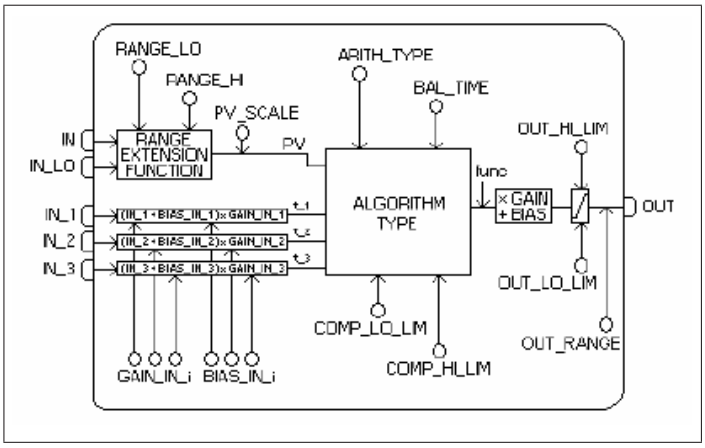

Figura 4.16: Esquemático do Bloco Aritmético

Os ganhos e *BIAS* associados a entradas e saídas permitem que este bloco se comporte como um neurônio artificial, bastando para isso usar seu algoritmo interno como um somador das entradas. Logo, faz-se possível a criação de neurônio computacional com três entradas. O algoritmo interno pode ser ajustado através do parâmetro *ARTH_TYPE*. Além disso, o bloco permite a criação de faixas de operação com limitações para entradas e saídas.

Bloco Caracterizador de Sinais (Signal Characterizer)

Este bloco serve para descrever funções genéricas quaisquer, possuindo duas seções internas, ele pode gerar duas saídas distintas a partir de suas entradas. A sua saída é a

resposta de uma função definida por duas tabelas internas (par x,y) contendo 20 pontos cada. Este bloco interpola os pontos e, baseado nessa interpolação, gera a resposta da função com relação a determinada entrada [Fie 2001]. A figura 4.17 mostra o esquemático do bloco [Smar n.d.].

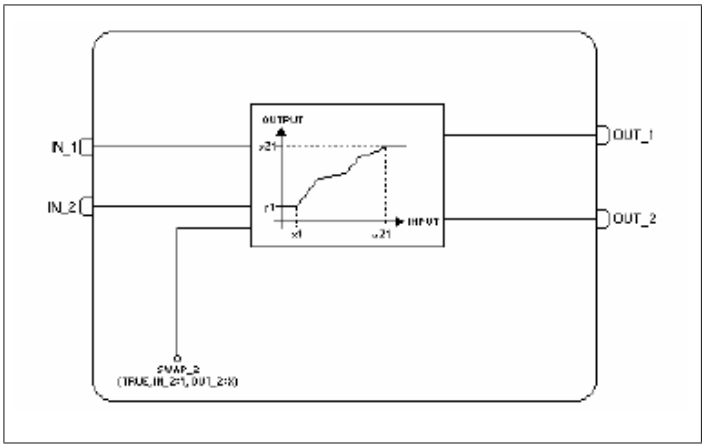

Figura 4.17: Esquemático do Bloco Caracterizador de Sinais

Uma característica importante presente no bloco é a condição de "bypassar" o valor da entrada através do parâmetro *BYPASS*.

A forma do bloco mapear o valor da saída, a partir de uma entrada específica baseando-se em uma tabela, faz com que o mesmo possa simular a função de ativação presente na arquitetura da RNA.

Como comentado no capítulo 3, em [Silva 2005] a estratégia usada para mapear as funções de ativação é aproximá-las à quantidade de pontos da tabela usando um algoritmo genético.

Bloco de Saída Analógica (Analog Output)

O bloco de saída analógica é um bloco funcional usado pelos equipamentos que trabalham como elementos de saída em um *loop* de controle, como válvulas, atuadores e posicionadores. O bloco recebe um sinal de outro bloco funcional e passa seu resultado para um transdutor de saída através de um canal interno de referência [Fie 2001].

A figura 4.18 apresenta sua estrutura interna [Smar n.d.].

O parâmetro *CHANNEL* corresponde ao canal do transdutor utilizado na saída. A escala *PV_SCALE* corresponde a faixa de trabalho da variável de processo. Já a *XD_SCALE*, corresponde a faixa de trabalho do dispositivo de saída.

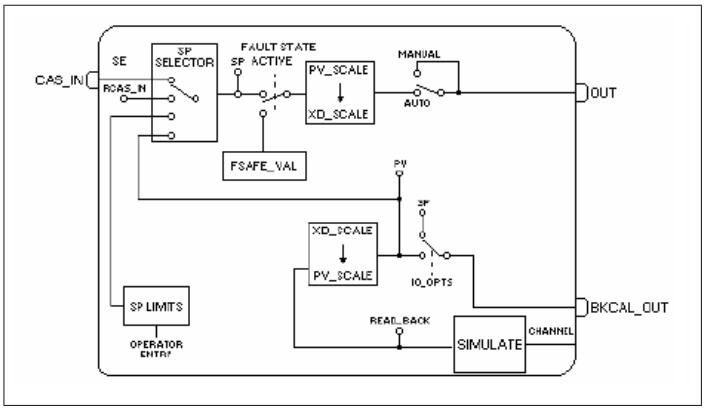

Figura 4.18: Esquemático do Bloco de Saída Analógica

Bloco de Controle PID (PID Control)

O bloco de controle PID oferece alguns algoritmos de controle que usam os termos proporcional, integral e derivativo. A figura 4.19 apresenta sua estrutura interna [Smar n.d.].

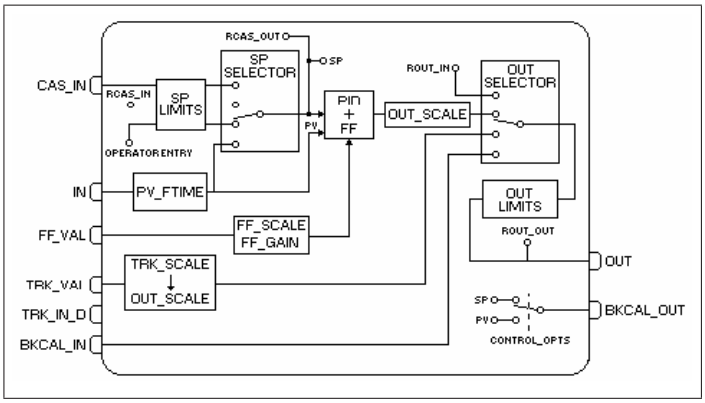

Figura 4.19: Esquemático do Bloco de Controle PID

Os parâmetros *GAIN* (Kp), *RESET* (Tr), e *RATE* (Td) são as constantes de sintonia para os termos P, I e D, respectivamente. Ganho é um número admensional. *RESET* e *RATE* são constantes de tempo expressas em segundos.

CAPÍTULO 5

Resultados e Discussões

5.1 Introdução

Neste capítulo são apresentados os testes para validação da proposta deste trabalho. Todos os experimentos usam como base a planta didática detalhada no capítulo 4.

Serão demonstrados dois cenários reais com a utilização da reconfiguração para ajuste da implementação feita nos blocos funcionais da rede FF.

O primeiro cenário consiste em um simples exemplo numérico para validação da solução proposta. Duas funções matemáticas são simuladas através de redes neurais artificiais e implementadas nos dispositivos da rede industrial. A reconfiguração da arquitetura da rede e, consequentemente, alteração da função utilizada será usada como experimento para testes de tempo de resposta do sistema e segurança vinculada ao processo.

O segundo cenário proposto baseia-se em duas alternativas para controle do processo simulado detalhado no anexo A. A dinâmica do processo utilizado sugere uma alternativa combinada ao controle clássico PID usado comumente na indústria. São desenvolvidos, então, dois escalonadores para ponderar os ganhos de dois controladores PI indicados para uma determinada seção do sistema de nível simulado. Um deles é baseado em Lógica *Fuzzy* e o outro utiliza a ferramenta da RNA. A reconfiguração do sistema será útil para validar a proposta em um ambiente com uma aplicação real de controle distribuído.

A construção das redes neurais para todos os experimentos seguiu o seguinte procedimento: primeiramente foram implementadas e treinadas em computadores PC convencionais (sem aproximações) a fim de se extrair os pesos dos neurônios e os sinais de *bias*. Logo em seguida, tiveram seus resultados avaliados de acordo com o erro médio quadrático entre o vetor de saída da rede neural e o conjunto de dados desejados, ainda em um ambiente simulado (*software* Matlab). Por fim, essa validação também foi realizada no ambiente real (através de estratégias implementadas nos nós da rede industrial).

A equação que retorna o valor do erro médio quadrático é indicada abaixo:

$$erro = \frac{1}{n}\sum_{i=1}^{n}(s_i - d_i)^2 \tag{5.1}$$

Sendo “n” o tamanho do vetor de saída, “s” a saída da rede neural e “d” a saída desejada.

5.2 Alteração entre Funções Matemáticas

Foram simuladas duas situações distintas para implementação das funções matemáticas a partir de redes neurais artificiais no ambiente FF.

A primeira implementação simula uma onda triangular gerada entre o período de 0 a 2Π. Fazendo uma associação com um processo industrial, a onda poderia implicar uma entrada de um determinado sistema físico. Contudo, a arquitetura escolhida e o treinamento realizado não geraram um resultado adequado para a simulação da onda triangular.

A arquitetura escolhida é de uma MLP 1-3-2-1 (uma entrada correspondente ao período, 3 neurônios na primeira camada oculta, 2 neurônios na segunda camada oculta e uma saída correspondente ao valor da função).

A figura 5.1 apresenta a validação da rede neural que simula a onda triangular no ambiente simulado. O erro médio quadrático encontrado foi de $3{,}5059 * 10^{-4}$.

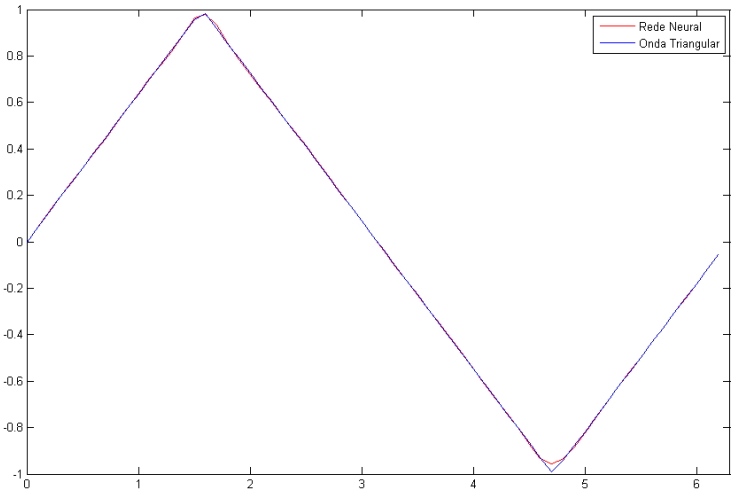

Figura 5.1: Validação da onda triangular no ambiente simulado

Já a figura 5.2 apresenta a validação da rede neural que simula a onda triangular no ambiente real da rede industrial. O erro médio quadrático encontrado foi de $4{,}2861 * 10^{-3}$.

A diferença entre os resultados obtidos nos dois ambientes se dá principalmente pela aproximação obtida das funções de ativação. Contudo, tal diferença pode ser contornada ao se utilizar vários blocos caracterizadores em cascata, aproximando a implementação das funções de ativação dos valores reais.

O outro teste baseia-se na simulação da função seno. No caso, uma rede MLP 1-3-1 (uma entrada, três neurônios na camada oculta e uma saída) foi utilizada para aproximar tal função. Fazendo a mesma associação com um processo industrial, a função seno, cujo treinamento obteve um resultado mais próximo da função real, poderia se tornar uma entrada mais suave (comparada ao primeiro caso) para um determinado sistema físico.

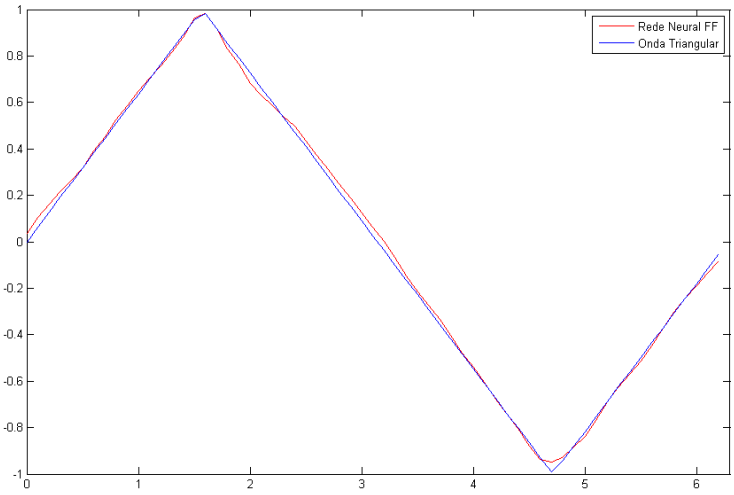

Figura 5.2: Validação da onda triangular no ambiente FF

As figuras 5.3 e 5.4 correspondem as validações realizada no ambiente simulado e na rede industrial FF, respectivamente. Os erros médios quadráticos foram de $2{,}3363 * 10^{-6}$ para o ambiente simulado e $1{,}0014 * 10^{-5}$ para a rede configurada nos instrumentos.

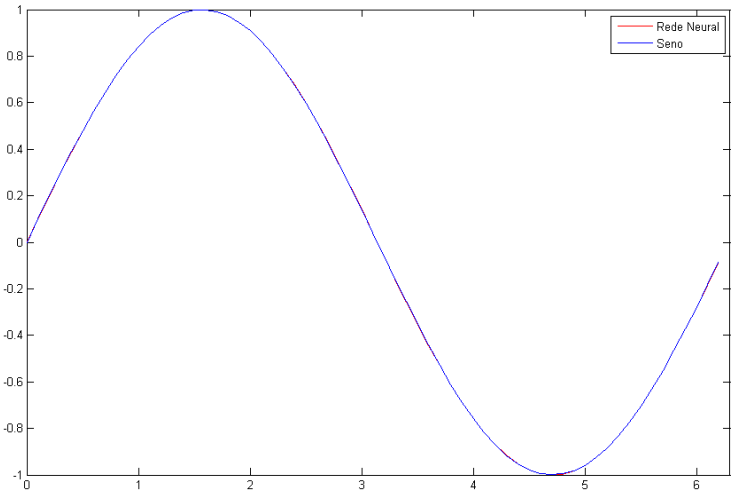

Figura 5.3: Validação da função seno no ambiente simulado

Para realizar a reconfiguração do sistema baseado nas duas opções acima, faz-se necessário uma implementação base nos dispositivos da rede, a qual seja capaz de as-

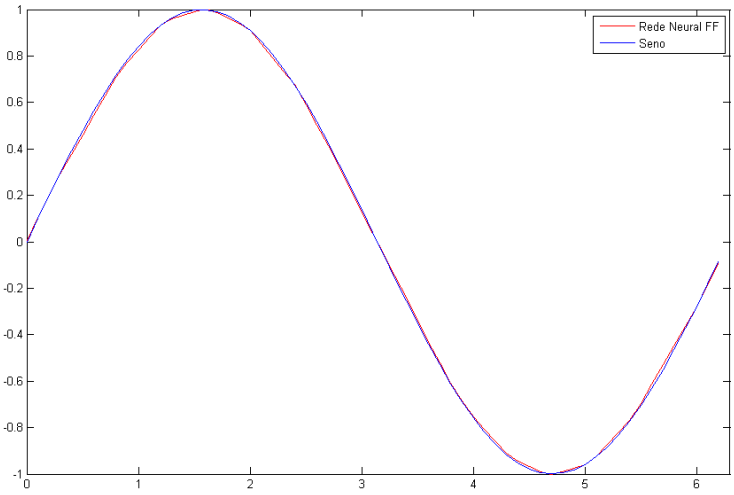

Figura 5.4: Validação da função seno no ambiente FF

sumir qualquer uma das duas arquiteturas de rede neural citadas nas simulações.

Logo, a rede neural apresentada no capítulo 3 (figura 3.5) foi instanciada na rede industrial didática com os blocos funcionais sendo distribuídos no elemento de entrada (IF302), na DFI302 e no dispositivo de saída (FI302). Como mostrado anteriormente, tal configuração é baseada em um modelo de arquitetura 3-3-3-3, sendo capaz de assumir o papel das duas redes citadas acima.

5.2.1 Sistema de Reconfiguração em Labview

Para realizar a reconfiguração da rede neural instanciada na rede FF através dos blocos funcionais, deve-se utilizar um cliente OPC que faça o envio dos parâmetros corretos da rede (pesos sinápticos e *bias*), assim como todos os ajustes necessários para adequar a nova arquitetura solicitada.

Baseado no sistema desenvolvido em [Cagni 2007], em que há o desenvolvimento de um sistema supervisório que configura arquiteturas de redes neurais em um DSP baseado no protocolo *Modbus*, realizou-se a implementação de um sistema coerente com a rede neural apresentada no capítulo 3 (figura 3.5). A *interface* em *Labview* pode ser vista na figura 5.5.

Esse sistema permite a adequação da arquitetura a partir da seleção da quantidade de neurônios artificiais por cada camada, atualização de pesos e *bias* dos nós ativados, assim como seleção da função de ativação presente nas três possíveis camadas.

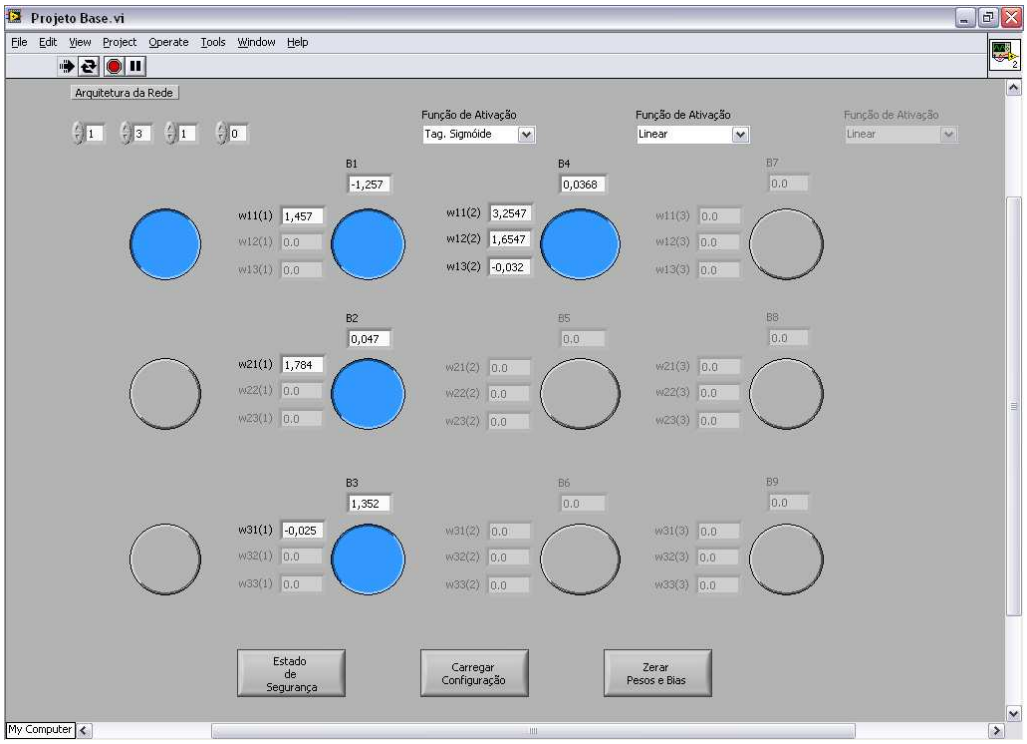

Figura 5.5: Exemplo de Interface em Labview para Reconfiguração da rede FF

A associação das variáveis da *interface* no *Labview* com as *tags* criadas pelo servidor OPC faz com que os blocos funcionais sejam atualizados nos dispositivos.

Cada peso e *bias* é enviado para a *tag* correspondente nos blocos aritméticos. A escolha da função de ativação carrega os ítens, previamente selecionados, na tabela dos blocos caracterizadores e a partir da arquitetura, há a ativação ou anulação dos *links* envolvidos no processo.

5.2.2 Levantamento e Análise dos Resultados

O primeiro teste realizado consistiu na reconfiguração da rede industrial da primeira simulação para a segunda (alteração da onda triangular para a função seno). As duas redes neurais criadas simulam funções que possuem valores entre -1 e 1 como resposta. Contudo, como observado na figura 5.6, tais valores eram extrapolados no momento da reconfiguração, visto que a rede industrial não interrompia seu funcionamento à medida que os novos valores de pesos e *bias* eram instanciados na estratégia.

Tal condição apresenta um risco em se tratando de sistemas reais, pois a saída da rede pode estar associada a um atuador ligado diretamente a um processo. Logo, faz-

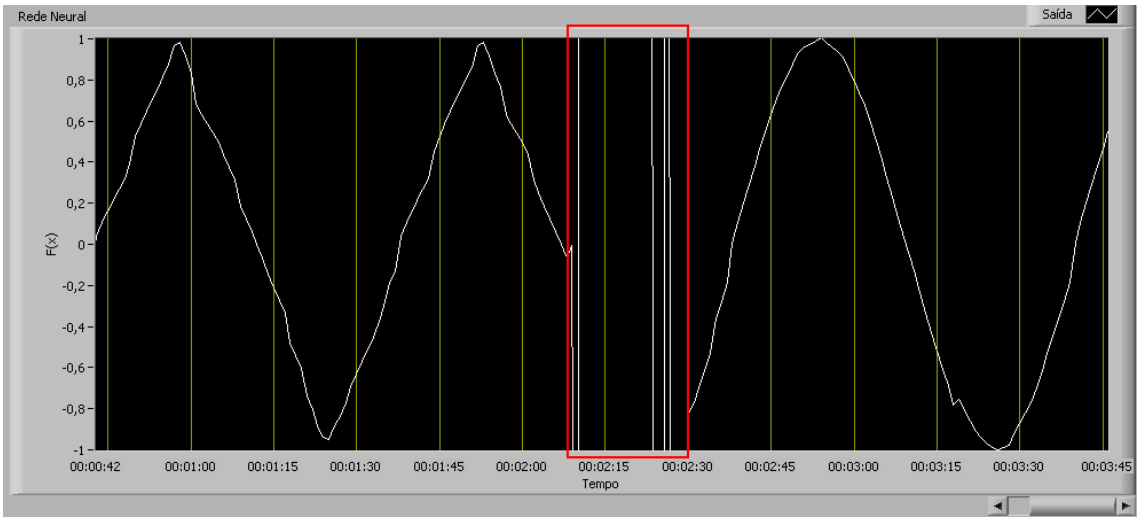

Figura 5.6: Tempo de resposta do sistema para reconfiguração entre as funções

se necessário estabelecer um critério de segurança para garantir um estado seguro no momento em que o sistema inicia o processo de reconfiguração.

De toda forma, vale ressaltar o tempo de resposta do sistema para a reconfiguração mencionada. Como pode ser observado na figura 5.6, o ajuste dos ganhos e *bias* da rede e a alteração dos *links* lógicos leva em torno de 23 segundos para acontecer.

Para acrescentar o estado de segurança antes de realizar qualquer tipo de configuração, faz-se necessário avaliar em que momento seria possível desabilitar tal estado para que a saída da rede neural volte a exercer o papel de saída da rede industrial.

Detalhando melhor o procedimento, no momento em que a reconfiguração é iniciada, o bloco funcional que contem a saída da rede neural deve alterar seu modo de execução para um modo manual e um valor de segurança deve ser mantido até que todos os pesos, *bias* e *links* sejam alterados. Com a operação concluída, é possível reajustar o modo de operação do último bloco da rede neural.

Para detectar o momento exato em que todos os valores de pesos e *bias*, assim como os ajustes nos *links*, fossem enviados, pretendia-se inicialmente analisar o tráfego da rede através do aplicativo *FBView* da SMAR.

O programa *FBView* (figura 5.7) permite avaliar o tráfego na rede industrial, pois o mesmo pode se conectar a DFI302 e configurá-la como *sniffer* na rede. Logo, seria possível analisar os pacotes e avaliar em que momento cada variável era enviada pela rede industrial.

Contudo, segundo [Zerbetto 2007] e [Smar n.d.], a DFI302 não consegue atuar de maneira ótima como *sniffer* quando já possui uma outra função na rede. Para o caso da

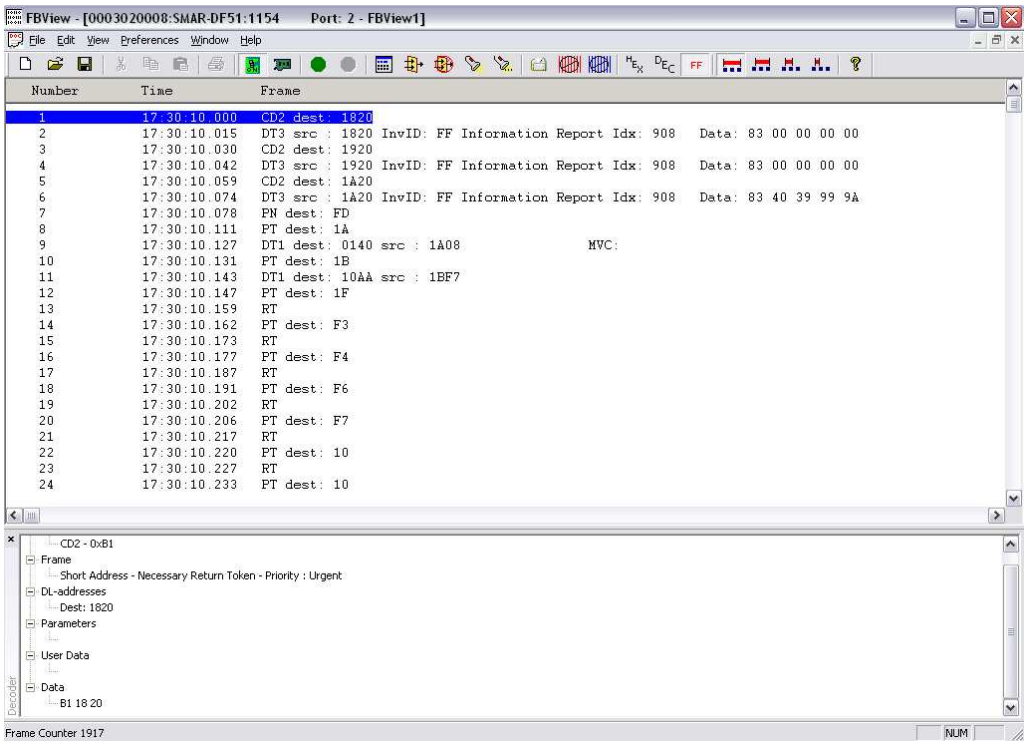

Figura 5.7: Tela do aplicativo FBView

rede didática, a DFI302 já atua como mestre do barramento, sendo necessário inserir outro dispositivo com a mesma capacidade de filtrar as mensagens que trafegam na rede.

Diante da impossibilidade de contar com essa alternativa, foi utilizado um esquema de redundância para análise do envio dos dados através do sistema em *LabView*. O cliente OPC ao enviar cada argumento da rede neural, realiza a leitura do mesmo item para que os valores sejam comparados até que todos os parâmetros estejam reconfigurados.

A figura 5.8 apresenta o tempo de resposta do sistema para alteração da função matemática. O quadro indicado na figura mostra que o experimento praticamente dobra o tempo de reconfiguração quando comparado ao primeiro teste sem a redundância no cliente OPC (o intervalo de alteração dos parâmetros da rede neural se aproxima dos 45 segundos). Contudo, para um melhor funcionamento do sistema, tal condição de segurança é necessária.

Depois de reconfigurada, a rede neural apresenta o comportamento da função seno como demonstrado na figura 5.9.

Para ser usado como parâmetro de comparação, a figura 5.10 apresenta a janela de *download* do programa *Syscon* para o envio da estratégia apresentada na figura 3.5 para os dispositivos da rede. Observa-se que o tempo de configuração dos dispositivos gira em torno de 9 minutos.

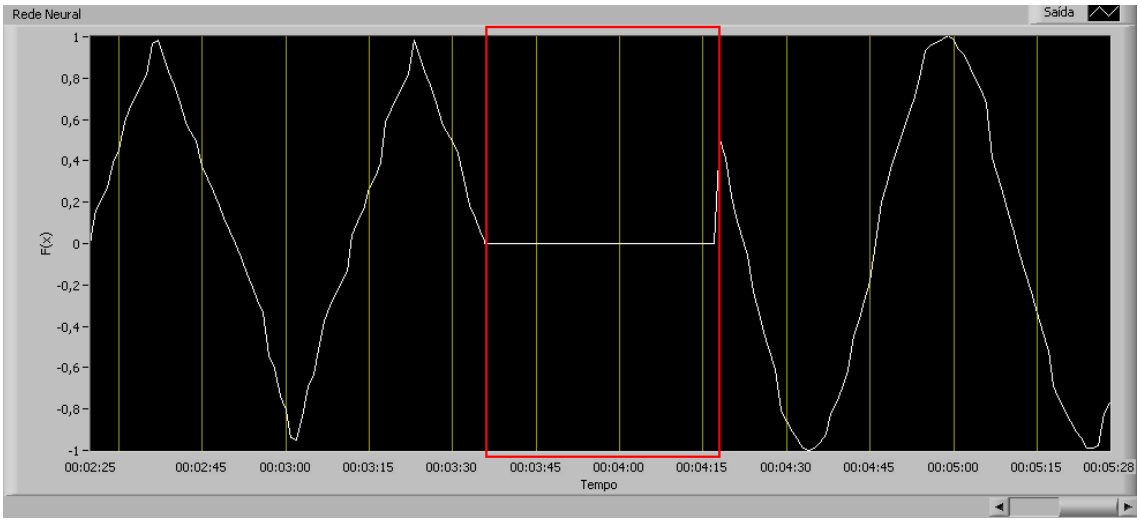

Figura 5.8: Tempo de resposta do sistema para reconfiguração com estado de segurança

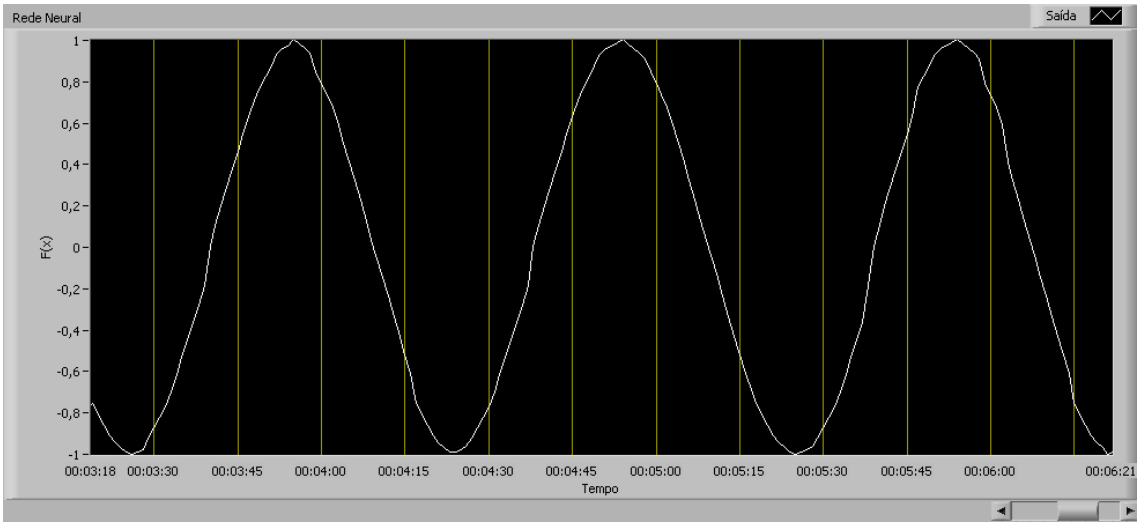

Figura 5.9: Resposta de rede neural simulando uma função seno no ambiente FF

5.3 Controle de um Tanque com Dinâmica Não-Linear

O segundo cenário para validação da proposta consiste em realizar a reconfiguração entre duas estratégias de controle distribuído. A figura A.1, a qual pode ser vista no anexo A, apresenta o modelo de um tanque simulado composto por três porções distintas. A equação A.2 é utilizada para demonstrar o comportamento da seção intermediária, em que a capacitância do tanque (área da seção) varia com o nível.

A figura 5.11 apresenta a *interface* da simulação do processo em *Labview*. A simu-

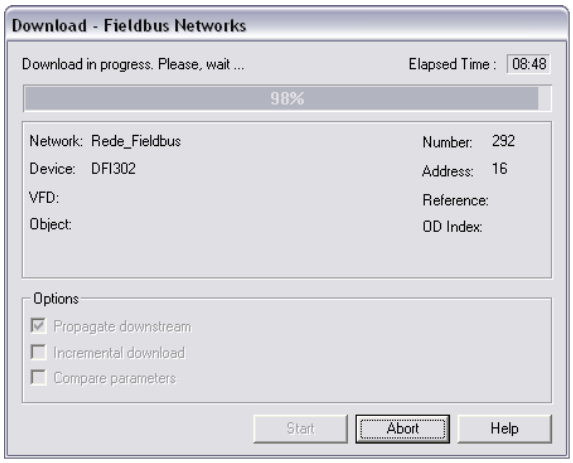

Figura 5.10: Janela de download do aplicativo Syscon

lação foi integrada a planta didática para os testes através das placas de aquisição de dados citadas no capítulo 4.

O nível do tanque é enviado através da placa D/A diretamente para o IF302 através de um *loop* de corrente de 4-20mA. Já a vazão de entrada (sinal de controle) é repassado da rede industrial através do FI302 em um sinal de 4-20mA. Tal sinal é convertido no equivalente de tensão de 0 a 5 *volts* e enviado ao conversor A/D para que seja usado pelo processo.

Em [Lima 2004] são propostos dois controladores PI fixos para as seções cilíndricas. As equações 5.2 e 5.3 apresentam as duas funções de transferência associadas a cada controlador, em que o parâmetro "E" corresponde ao sinal de erro entre o *setpoint* de cada controlador e saída da planta.

Para a parte inferior do tanque:

$$PI_1(s) = \frac{U_1(s)}{E_1(s)} = 1[1 + \frac{1}{10s}] \tag{5.2}$$

Para a parte superior do tanque:

$$PI_2(s) = \frac{U_2(s)}{E_2(s)} = 4[1 + \frac{1}{40s}] \tag{5.3}$$

5.3.1 "Design" das Aplicações em Blocos Funcionais

As duas estratégias de controle distribuído nos dispositivos da rede *Foundation Fieldbus* são baseadas em escalonadores para os controladores PID citados acima. Nos dois

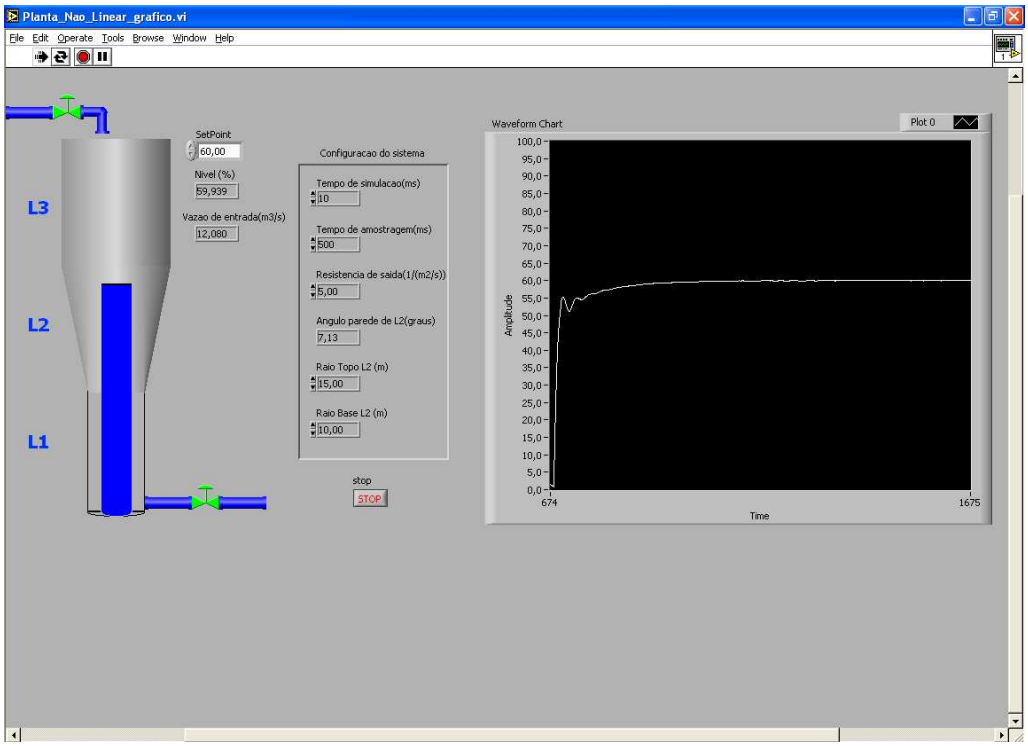

Figura 5.11: Interface do processo simulado em Labview

projetos, a soma das ações dos controladores ponderadas serve como sinal de controle do processo como um todo. A diferença, no caso, está associada a maneira como os escalonadores são implementados.

O primeiro escalonador dos controladores PID é baseado na Lógica *Fuzzy*. As funções de pertinência *fuzzy* que ponderam as ações dos controladores PI fixos são apresentadas na figura 5.12. Assim, a ação de controle, a ser aplicada no processo, é a média ponderada das ações de controle geradas pelos controladores PI fixos [Lima 2004].

A implementação das funções de pertinência pode ser realizada através de blocos caracterizadores, os quais simulam o comportamento de uma determinada função a partir da especificação de 20 pares ordenados (x, y).

O segundo escalonador é baseado em uma rede neural, a qual segue o comportamento da equação A.2 para realizar o escalonamento dos controladores, visto que o comportamento da seção cônica é não-linear [Silva 2005]. A figura 5.13 apresenta as funções de ponderação dos controladores PID, as quais seguem a equação anteriormente citada.

Esse escalonador pode ser implementado na rede industrial através de uma rede neural, a qual simula seu comportamento. A figura 5.14 apresenta a validação da rede treinada para exercer o comportamento do escalonador citado. O erro médio quadrático obtido é

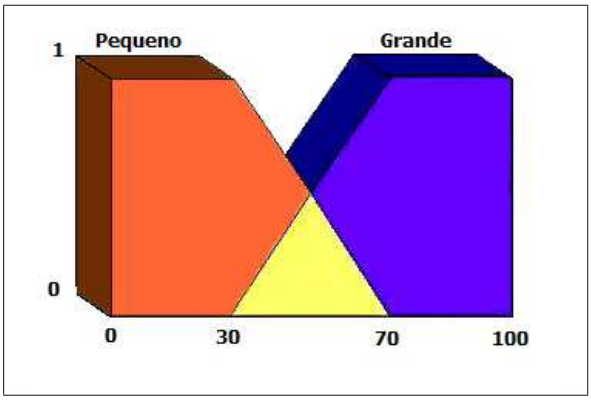

Figura 5.12: Funções de pertinência para o modelo não-linear do processo

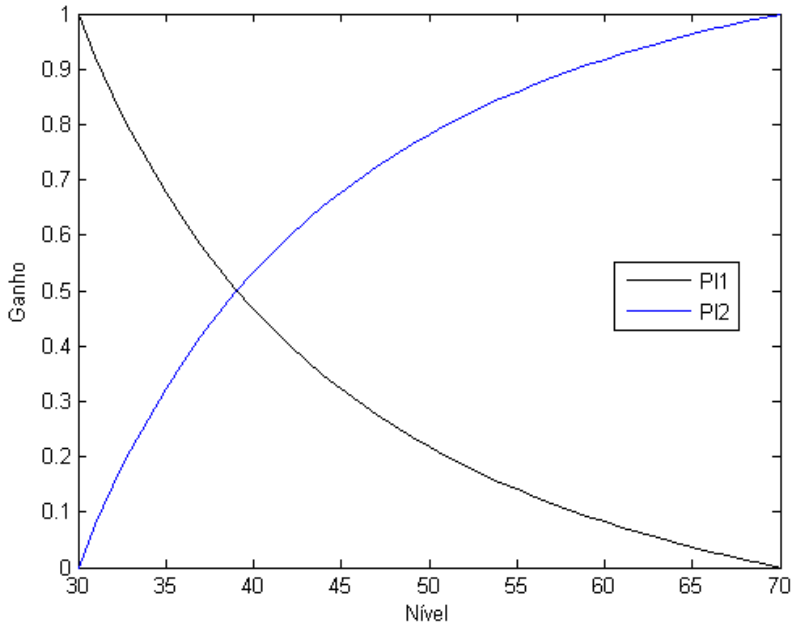

Figura 5.13: Funções de ponderação dos controladores PID

de $3{,}8282 * 10^{-5}$.

A pré-configuração dos dois escalonadores de forma simultânea na rede industrial permite que, a partir de uma necessidade futura, haja a reconfiguração da rede FF para atender a uma das duas estratégias selecionadas. A figura 5.15 apresenta o *design* da aplicação contendo os dois tipos de escalonadores.

Como pode ser observado na figura, o nível do tanque é enviado para os controladores PID, para a rede neural e para os blocos que processam a função de pertinência *fuzzy*.

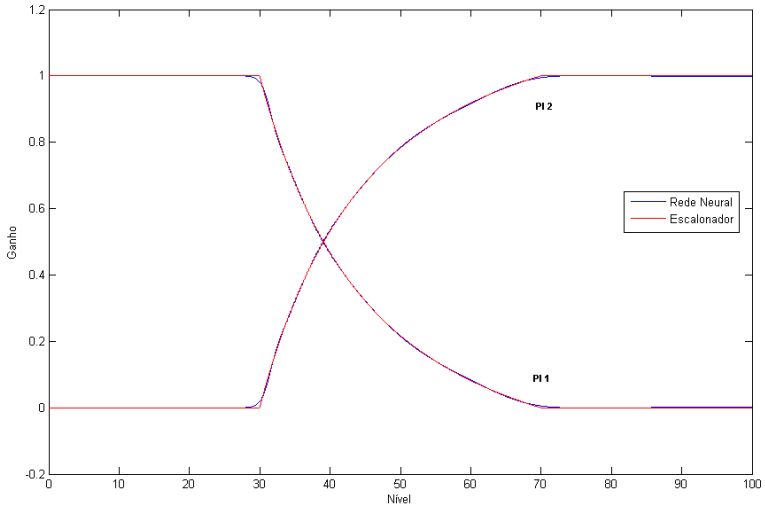

Figura 5.14: Rede neural que simula as funções de ponderação dos controladores PID

Os controladores realizam o cálculo do algoritmo PID e tem como saída os sinais de controle. Tais sinais podem ser ponderados tanto pelos escalonadores *fuzzy*, quanto pelo escalonador neural. A ativação dos *links* lógicos é o fator preponderante para a escolha do tipo de controlador.

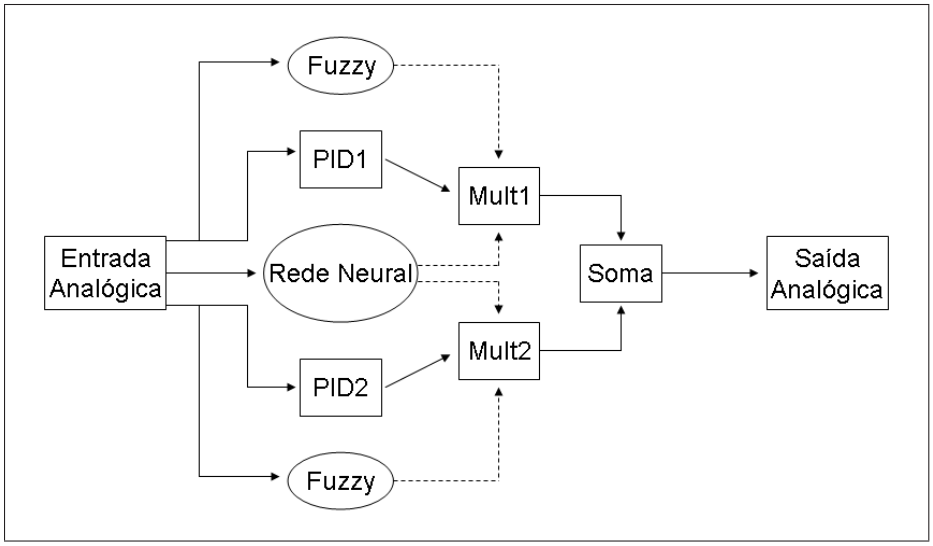

Figura 5.15: Design da aplicação para implementação dos dois escalonadores (Fuzzy e Neural)

A figura 5.16 apresenta a implementação dos escalonadores em blocos funcionais.

Os nós da rede pintados de azul correspondem as funções de pertinência *fuzzy*. Os nós pintados de amarelo são os blocos funcionais que implementam a rede neural. Também estão destacados na figura os dois controladores PID, os quais são representados pelos blocos pintados de verde.

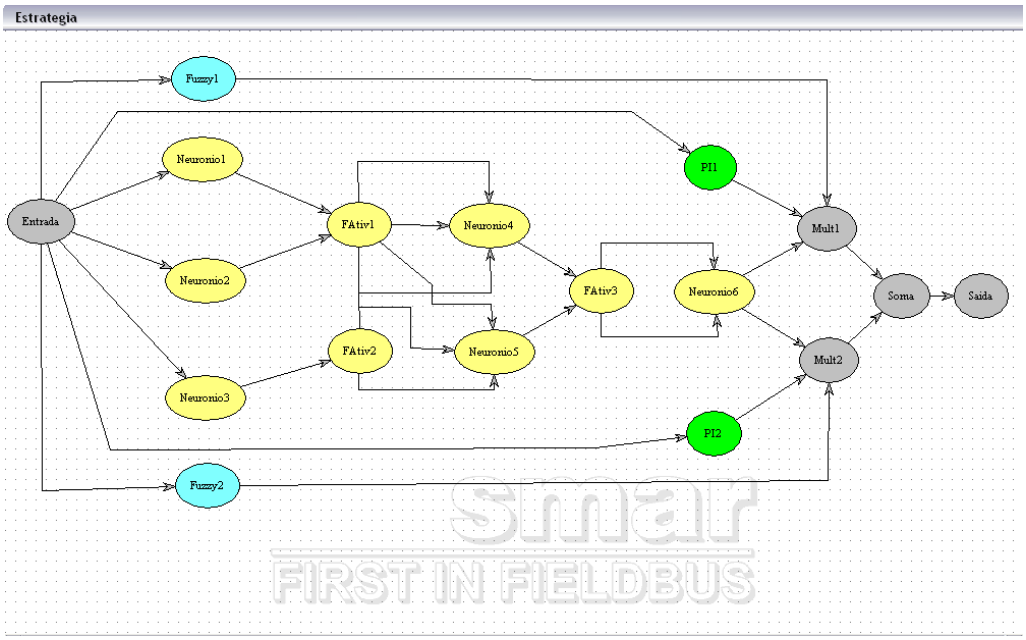

Figura 5.16: Implementação em blocos funcionais para os dois escalonadores

5.3.2 Levantamento e Análise dos Resultados

Os testes realizados partiram de uma pré-configuração, na qual o escalonador com funções de pertinência *fuzzy* estava ativado na rede e foi feita a reconfiguração para que o escalonador neural atuasse nos dispositivos.

Comparado com os testes do primeiro cenário, o modo de realizar a reconfiguração da rede é semelhante. No caso, ao invés de realizar a troca de arquitetura da rede neural, para essa implementação, faz-se necessário o ajuste dos parâmetros de ganho dos blocos funcionais aritméticos que fazem o papel dos multiplicadores na estratégia.

Cada bloco aritmético possui três entradas físicas que podem ser ponderadas por constantes definidas nos blocos. Através de um cliente OPC, é possível realizar a atualização desses valores e ativar o *link* lógico do escalonador *fuzzy* ou do escalonador neural.

A figura 5.17 apresenta o tempo de resposta do sistema para realizar a reconfiguração do escalonador *fuzzy* para o escalonador neural para uma mesma faixa de *setpoint* (60%

do nível do tanque). O início do quadro indica o momento em que o sinal de controle é anulado e o nível do tanque começa a cair. Nesse instante, os parâmetros dos blocos aritméticos são ajustados para ativar o escalonador neural.

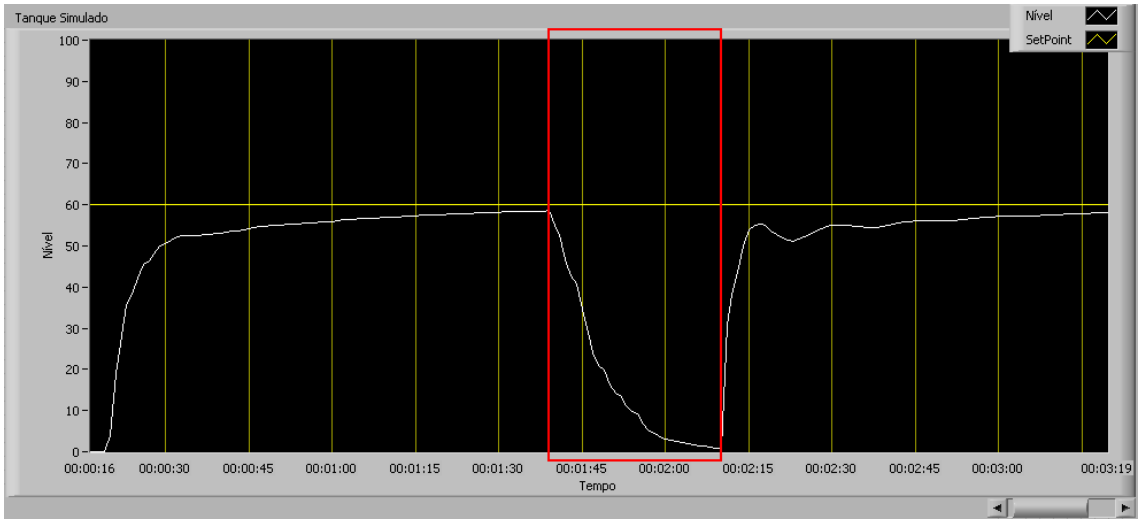

Figura 5.17: Tempo de resposta do sistema para reconfiguração com setpoint em 60%

O parâmetro para reativar o sinal de controle também foi a redundância presente no cliente OPC, o qual espera a confirmação de leitura de todas as variáveis enviadas para reajustar o modo de operação do bloco de saída analógica. Como pode ser visto na figura 5.17, o tempo total de reconfiguração girou em torno de 30 segundos.

A diferença existente entre os tempos de reconfiguração do primeiro e do segundo cenário, deve-se ao fato de que, no primeiro caso, toda rede neural deve ser alterada, consumindo um pouco mais de tempo para confirmação da escrita em todos os parâmetros da rede.

A figura 5.18 apresenta outro teste realizado. Neste exemplo, durante a anulação do sinal de controle para realizar a reconfiguração da estratégia de controle, altera-se o valor do *setpoint* dos controladores de 20% para 80% do nível do tanque.

O tempo de resposta do sistema é semelhante ao do primeiro exemplo, aproximando-se dos 30 segundos. No entanto, esse exemplo prático se aproxima mais de uma situação real, em que as diferente faixas de operação do operação do processo podem exigir diferentes funcionalidades agregadas ao sistema de controle.

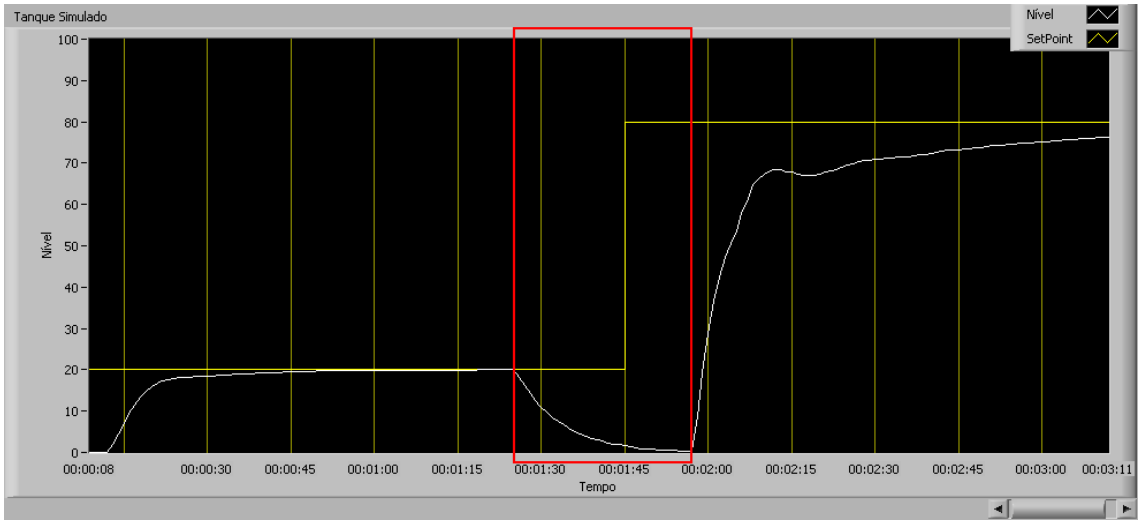

Figura 5.18: Tempo de resposta do sistema para reconfiguração com setpoint entre 20% e 80%

CAPÍTULO 6

Considerações Finais

6.1 Conclusões

A presente dissertação apresentou um estudo da viabilidade técnica de se realizar reconfigurações nas estratégias distribuídas em dispositivos de campo que seguem o padrão de comunicação *Foundation Fieldbus*, mostrando a possibilidade de se realizar tal implementação com garantias em questões temporais e questões de segurança.

Alguns aspectos da tecnologia de redes industriais em questão, como a interoperabilidade, foram mantidos, pois as implementações tiveram como base os blocos funcionais padronizados pelo protocolo.

Outros aspectos como a flexibilidade e a adaptabilidade foram fortalecidos com a possibilidade de se estender a gama de funciolidades da rede industrial através da combinação de poderosas ferramentas da inteligência artificial, como as redes neurais, e um modo prático de configurá-las em tempo de execução.

6.2 Contribuições da Dissertação

Além da relevância já comentada de um meio para realizar a reconfiguração dos instrumentos de uma rede *Foundation Fieldbus* em tempo de execução de suas tarefas, vale ressaltar também as demais contribuições obtidas com o desenvolvimento do trabalho.

Uma delas consiste na criação de um ambiente híbrido para testes diversos relativos a uma rede industrial. O meio criado para realizar a comunicação de um processo simulado com um ambiente de rede industrial real tem grande valor no aspecto técnico, pois torna-se viável o estudo de diversas aplicações nas mais variadas áreas de automação de processos, desde que o processos simulados sigam fielmente a caracterização dos processos reais.

As demais contribuições estão associadas ao âmbito acadêmico com algumas publicações feitas em conjunto com demais pesquisadores do Laboratório de Avaliação de Medição em Petróleo (LAMP). Tais publicações foram citadas no decorrer e correspondem aos estudos feitos para criação de uma arquitetura multi-agente nos dispositivos de campo da rede industrial [Machado et al. 2008*a*] [Machado et al. 2008*b*] (o modo de reconfiguração das estratégias no ambiente de rede é base para tal estudo), assim como a criação do ambiente híbrido para concepção de sensores de *software* [Costa et al. 2008] [Rodrigues et al. 2008].

6.3 Perspectivas e Trabalhos Futuros

Para uma avaliação ainda mais satisfatória do modo de reconfiguração das estratégias distribuídas em dispositivos de uma rede industrial *Foundation Fieldbus*, faz-se necessário a incorporação desta metodologia a processos mais complexos da industria petroquímica.

Como citado no capítulo 4, o ambiente híbrido pode ser associado ao *software* de simulação de processos químicos *Aspen Hysys*. Com tal integração, é viável realizar o estudo de diferentes sensores virtuais aplicados aos processos da indústria petroquímica e utilizar o modo de reconfiguração de estratégias distribuídas para agregar um nível de automação ainda maior ao sistema.

Além disso, destaca-se também como trabalho futuro a continuação da concepção de uma arquitetura multi-agente em um ambiente de rede industrial *Foundation Fieldbus*.

Essa arquitetura prevê a criação de alguns níveis de agentes inteligentes no barramento de campo responsáveis por tarefas como diagnóstico de falhas, filtragem de sinais, dentre outras. Tais agentes serão implementados a partir de diferentes estratégias concebidas através do blocos funcionais e a reconfiguração das estratégias estará associada a ativação dos diferentes agentes. No caso, tal trabalho vem sendo desenvolvido em parceria com o aluno de doutorado do Programa de Pós-graduação de Engenharia Elétrica e Computação da UFRN, Vinícius Machado.

Referências Bibliográficas

Advantech (n.d.), 'Home Page da Advantech. http://www.advantech.com/. Acessada em Dezembro/2007'.

Berge, Jonas (2001), *Fieldbuses for Process Control: Engineering, Operation and Maintenance*, ISA - The Instrumentation, Systems and Automation Society.

Brandão, Dennis (2005), Ferramenta de Simulação para Projeto, Avaliação e Ensino de Redes Fieldbus, Tese de doutorado, Universidade de São Paulo.

Brennan, Robert, Martin Fletcher & Douglas Norrie (2002), 'An agent-based approach to reconfiguration of real-time distributed control systems', *IEEE Transactions on Robotics and Automation* **18**(4), 444–451.

Buse, D. P. & Q. H. Wu (2004), 'Mobile agents for remote control of distributed systems', *IEEE Transactions on Industrial Eletronics* **51**(6), 1142 – 1149.

Cagni, Eloi (2007), Software inteligente embarcado aplicado à correção de erro na medição de vazão em gás natural, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.

Cagni, Eloi, David Pereira, Adrião Dória, Jorge Melo & Luiz Oliveira (2005), 'The implementation of the self-calibration, self-compensation and self-validation algorithms for foundation fieldbus sensors are presented using standard function blocks', *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications* **20**(22), 220–225.

Cicillini, Daniele (2007), Desenvolvimento de um algoritmo de escalonamento para rede foundation fieldbus, Dissertação de mestrado, Universidade de São Paulo.

Costa, Bruno, Leonardo Guanabara, Igor Rodrigues, Daniel Martins, Adrião Dória, Jorge Melo & Luiz Oliveira (2008), 'Ambiente híbrido para concepção de sensores de software aplicados aos problemas da indústria do petróleo', *Rio Oil & Gas 2008* .

Costa, Isabele (2006), Projeto e implementação em ambiente foundation fieldbus de filtragem estocástica baseada em análise de componentes independentes, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.

Fernandes, Raphaela (2007), Detecção e isolamento de falhas em sistemas dinâmicos baseados em redes neurais, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.

Fie (2001), *FOUNDATION Specifications - Function Block Application Process*.

Fie (2003), *FOUNDATION Specifications - SP-150 H1, HSE, and User Layer Technical Specifications*.

Fortuna, L., S. Graziani, Alessandro Rizzo & M. Xibilia (2006), *Soft Sensors for Monitoring and Control of Industrial Processes*, 1ª edição, Springer.

Fortuna, L., S. Graziani & M. Xibilia (2005), 'Virtual instruments in refineries: Data monitoring for environmental quality', *IEEE Instrumentation e Measurement Magazine* **8**(4), 26–34.

Foundation, Fieldbus (n.d.), 'Home Page do Fieldbus Foundation. http://www.fieldbus.org/. Acessada em Dezembro/2007'.

Haykin, Simon (2004), *Redes Neurais: Princípios e Prática*, Bookman.

Labview (n.d.), 'Home Page da National Instruments. http://www.ni.com/. Acessada em Setembro/2008'.

Lima, Fábio (2004), Estratégia de escalonamento de controladores pid baseado em regras fuzzy para redes industriais foundation fieldbus usando blocos padrões, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.

Machado, Vinícius, Adrião Duarte Dória, Jorge Dantas de Melo, Leonardo Ramalho & Juliana Medeiros (2008*a*), 'Multiagent architecture for function blocks: Intelligent configuration strategies allocation', *IEEE International Conference on Industrial Informatics* **13**(16), 1377–1382.

Machado, Vinícius, Adrião Duarte Dória, Jorge Dantas de Melo, Leonardo Ramalho & Juliana Medeiros (2008*b*), 'A neural network multiagent architecture applied to fieldbus intelligent control', *IEEE International Conference on Emerging Technologies and Factory Automation* pp. 567–574.

Marangoni, Cintia (2005), Implementação de uma Estratégia de Controle Distribuída em uma Coluna de Destilação, Tese de doutorado, Universidade Federal de Santa Catarina.

Pantoni, Rodrigo (2006), Desenvolvimento e implementação de uma descrição de dispositivos aberta e não-proprietária para equipamentos foundation fieldbus baseada em xml, Dissertação de mestrado, Universidade de São Paulo.

Prayati, Aggeliki, Christos Koulamas, Stavros Koubias & George Papadopoulos (2004), 'A methodology for the development of distributed real-time control applications with focus on task allocation in heterogeneous systems', *IEEE Transactions on Industrial Eletronics* **51**(6), 1194–1207.

Rodrigues, Igor, Bruno Costa, Leonardo Guanabara, Daniel Martins, Adrião Dória, Jorge Melo & Luiz Oliveira (2008), 'Ambiente para concepção de sensores de software em redes industriais foundation fieldbus', *VIII Industrial Applications International Conference - Induscon* .

Sam (2000), *Technical Information: Foundation Fieldbus*.

Silva, Diego (2005), Redes neurais artificiais no ambiente de redes industriais foundation fieldbus usando blocos funcionais padrões, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.

Smar, Equipamentos (n.d.), 'Home Page da Smar. http://www.smar.com.br/. Acessada em Novembro/2007'.

Strasser, T., I. Müller, C. Sünder, O. Hummer & H. Uhrmann (2006), 'Modeling of reconfiguration control applications based on the iec 61499 reference model for industrial process measurement and control systems', *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications* pp. 127–132.

Zanata, Diogo (2005), Desenvolvimento de um sensor virtual empregando redes neurais para medição da composição em uma coluna de destilação, Dissertação de mestrado, Universidade de São Paulo.

Zerbetto, Angelo (2007), Análise do impacto da comunicação em redes foundation fieldbus no desempenho de sistemas de controle, Dissertação de mestrado, Universidade Federal do Rio Grande do Sul.

APÊNDICE A

Modelo Não-Linear - Tanque Cônico

O modelo não-linear descrito abaixo corresponde a simulação criada para validação do sistema. Como pode ser visto na figura A.1, o tanque é formado por dois cilindros e uma parte cônica.

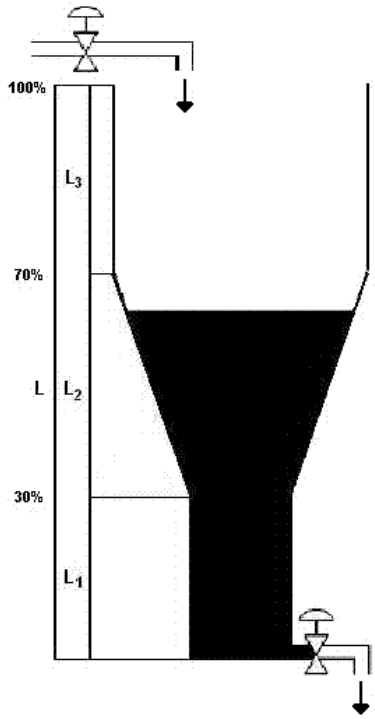

Figura A.1: Modelo físico não-linear simulado de um tanque

De acordo com [Lima 2004], a escolha de um sistema com controle de nível é interessante, visto que há uma grande tendência para sua utilização no meio acadêmico, além de ser bastante utilizado no meio industrial. O sistema em questão também é interessante por apresentar a integração de dois sistemas (linear e não-linear).

A função de transferência do tanque foi modelada através da equação A.1. O sinal de saída é o nível (m) e o sinal de entrada é a vazão da bomba (m^3/s).

$$G(s) = \frac{L(s)}{U(s)} = \frac{R}{RCs+1} \tag{A.1}$$

Em que:

- R → Resistência do orifício de saída;
- C → Capacitância do tanque (Área da seção);
- L → Nível do tanque;
- U → Vazão de entrada.

As partes cilíndricas (partes lineares) foram simuladas através de dois modelos matemáticos invariantes no tempo, alterando-se apenas os valores de C, conforme características posteriormente citadas. Para a simulação da parte cônica (parte não-linear), foi utilizado o mesmo modelo, fazendo com que fosse variante, no qual C varia de acordo com o nível (L) (segundo a equação A.2).

$$C(L) = (\frac{\pi}{3}) \cdot (3 \cdot K^2 \cdot L^2 + 4 \cdot r_{base} \cdot K \cdot L + 2 \cdot h \cdot K^2 \cdot L + r_{base} + 2 \cdot r_{base} \cdot K \cdot L) \tag{A.2}$$

Em que:

$$K = \frac{r_{topo} - r_{base}}{0.4} \tag{A.3}$$

$$h = \frac{r_{base} \cdot 0.4}{r_{topo} - r_{base}} \tag{A.4}$$

O tanque simulado possui as seguintes características:

- Altura total = 1m;
- Raio base (r_{base}) = 0.2m;
- Raio topo (r_{topo}) = 0.4m;
- Resistência do orifício de saída (R) = 80s/m^2;
- Altura do cilindro inferior = 0.3m;
- Altura do cilindro superior = 0.3m;