

# Implementação de Redes Neurais Artificiais em Ambientes de Redes Industriais *Foundation Fieldbus* com o uso de Blocos Funcionais Padrões

Diego R. C. Silva, Fábio S. Lima, Luiz Affonso Guedes, Adrião D. Dória Neto, Jorge D. Melo  
 Departamento de Engenharia de Computação e Automação  
 Universidade Federal do Rio Grande do Norte  
 Natal, Brasil

E-mail: diego@dca.ufrn.br, fabio@dca.ufrn.br, affonso@dca.ufrn.br, adriao@dca.ufrn.br, jdmelo@dca.ufrn.br

**Resumo**— Este artigo trata da implementação de redes neurais artificiais no ambiente de redes industriais *Foundation Fieldbus* (FF), através do uso de blocos funcionais padronizados pela *Fieldbus Foundation*. Isso possibilita a implementação de vasta gama de aplicações que envolvem esta ampla ferramenta matemática, como: controle inteligente, detecção de falhas, entre outras, em qualquer sistema padrão FF.

**Palavras-Chave**— Redes Neurais Artificiais, Redes Industriais e Inteligência Computacional.

## I. INTRODUÇÃO

A automação industrial tem evoluído bastante desde a década de 60. Com a popularização da microeletrônica, surgiram os primeiros CLPs (Controladores Lógico Programáveis). A partir destes, os projetos físicos foram evoluindo até chegar às redes para automação industrial. Finalmente, impulsionados pelo avanço da comunicação digital, chegou-se hoje ao que se chama de FCS (Field Controller System) [1]

Dentre os vários sistemas de controle de campo existentes no mercado, destaca-se o *Foundation Fieldbus* (FF) [1], por ser acompanhada por uma fundação que tem como principais objetivos a especificação de um padrão aberto, o que torna possível a interoperabilidade entre equipamentos de diferentes fabricantes. Um segundo ponto importante na arquitetura FF é a organização da camada de aplicação em blocos, que facilita o desenvolvimento de novas funcionalidades pelo fato dessas poderem ser encapsuladas, definindo apenas um interface de entrada e saída de sinais.

Existem alguns blocos de função padronizados pela FF, como controladores e algoritmos matemáticos por exemplo [2], porém faltam ainda funcionalidades avançadas nesse tipo de sistema, como lógica fuzzy, redes neurais, etc.

Algumas técnicas de inteligência computacional poderiam vir a ser úteis no controle e supervisão dos processos dos mais variados tipos, como por exemplo no controle inteligente [3], medição indireta [4], detecção de falhas [5]. Mas esses blocos “avançados” ainda não são padronizados pela fundação, o que deixa margens para alguns fabricantes implementarem à sua maneira, o que não garante a interoperabilidade.

O objetivo deste trabalho consiste em disponibilizar a ferramenta de redes neurais artificiais no ambiente *Foundation Fieldbus*, fazendo uso, para isso, apenas de blocos funcionais

padronizados pela fundação. O que torna a solução universal entre ambientes e dispositivos de quaisquer fabricantes, e possibilitando novas aplicações nos processos a serem controlados.

O restante do artigo está organizado da seguinte maneira: na Sessão 2, apresenta-se o padrão FF e suas particularidades, introduz-se o conceito de blocos funcionais e cita-se alguns exemplos dos mesmos. Na Sessão 3 descreve-se os conceitos fundamentais de redes neurais e as soluções utilizadas para sua implementação no ambiente de redes *Foundation Fieldbus*, usando blocos funcionais padronizados. Na Sessão 4 descreve-se o ambiente de testes e apresenta-se os resultados obtidos para dois experimentos. E, finalmente, na Sessão 5, encontra-se uma breve conclusão seguida de sugestões para trabalhos futuros.

## II. REDES *Foundation Fieldbus*

A *Fieldbus Foundation* [6] é uma organização independente sem fins lucrativos cujo propósito é desenvolver e manter um padrão internacionalmente uniforme de redes de campo para automação de processos, o *Foundation Fieldbus*. Uma de suas principais características é a organização da camada de aplicação em blocos, que por ser aberta e completamente especificada permite atingir um dos principais objetivos da fundação, a interoperabilidade.

Esses blocos, da camada de aplicação, são divididos em três grandes categorias: os blocos de recursos, os blocos transdutores e os blocos funcionais. Estes últimos são responsáveis pela manipulação dos sinais fornecidos pelos blocos de recurso, depois de tratados pelos blocos transdutores.

Cada bloco funcional, pode ser visto como um bloco de *software* com interfaces de entrada e saída bem definidas, que exerce processamentos de sinais, deixando transparente para o usuário os detalhes internos de implementação. Esse encapsulamento possibilita o desenvolvimento descentralizado e independente de novos blocos funcionais, o que torna simples a tarefa de suprir a rede com novas funcionalidades.

Outro ponto positivo da arquitetura baseada em blocos funcionais é a capacidade de se interligar os blocos a fim de combinar algoritmos de processamento em sinais de entrada de um dispositivo. Isso permite processamentos complexos e aumenta o número de aplicações possíveis.

Entre os blocos funcionais padronizados pela fundação pode-se citar alguns a seguir: AI (Entrada Analógica), AO (Saída Analógica), DI (Entrada Discreta), DO (Saída Analógica), PID (Controlador Proporcional-Integral-Derivativo), ARITHM (Aritmético), CHAR (Caracterizador), entre outros. Os dois últimos são usados neste trabalho e serão detalhados. A Figura 1 mostra o esquema interno do bloco aritmético padrão FF. Nela, pode-se perceber a existência de um sub bloco intitulado *Algorithm Type* que pode assumir diferentes configurações e é escolhido pelo usuário. As entradas IN, IN\_1, IN\_2, IN\_3, são entradas do bloco aritmético e OUT é a saída processada deste bloco.

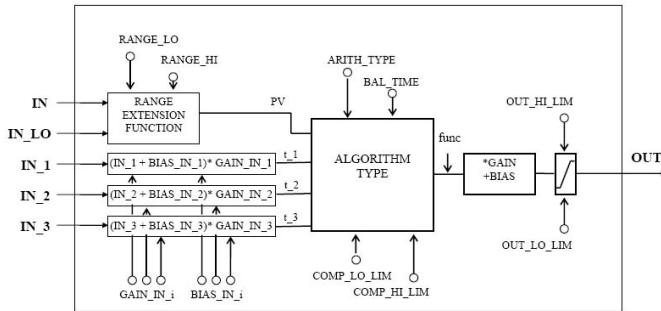


Fig. 1. Esquema interno do bloco Aritmético padrão FF.

As diversas funções assumidas por esse sub-bloco são listadas na tabela I e são mapeadas através do parâmetro ARITH\_TYPE.

TABELA I  
FUNCIONALIDADES DO BLOCO ARITMÉTICO

N.	Descrição do Algoritmo
1	Compensação de Fluxo, linear
2	Compensação de Fluxo, raiz quadrada
3	Compensação de Fluxo, aproximação
4	Fluxo BTU
5	Múltiplo Divisor Tradicional
6	Média
7	Somador Tradicional
8	Polinômio de Quarta Ordem
9	Compensador de Nível HTG

A Figura 2 ilustra o esquema interno do bloco caracterizador de funções.

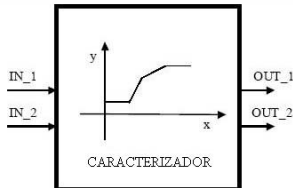


Fig. 2. Esquema interno do bloco caracterizador.

Como o próprio nome diz, este bloco tem o objetivo de caracterizar funções, ou seja, tenta-se construir a função desejada por meio da interligação de pontos, com a limitação de 21 pontos. Esses pontos, escolhidos pelo usuário, estão na

forma de parâmetros e são preenchidos através de um *software* configurador.

### III. REDES NEURAIS E SUA IMPLEMENTAÇÃO NO AMBIENTE FF

As redes neurais são formadas por células chamadas neurônios, e são responsáveis pelo processamento de informações no cérebro. As redes neurais artificiais, por sua vez, são formadas por conjuntos de neurônios artificiais que consistem de um modelo matemático de um neurônio natural. A Figura 3 mostra esse modelo [7].

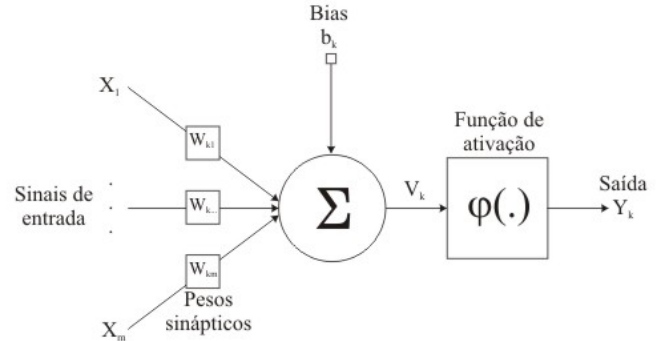


Fig. 3. Modelo matemático de um neurônio artificial.

Os neurônios são conectados através de interligações conhecidas por sinapses, ou pesos sinápticos. A cada peso sináptico é associado um ganho, que gera uma transformação no sinal de entrada. Todos os sinais de entrada são somados e um bias é aplicado no final afim de aumentar ou diminuir a entrada líquida da função de ativação, que é responsável pela não linearidade.

Existem duas características principais de estruturas de redes neurais: redes de alimentação direta e redes recorrentes [8]. Uma rede de alimentação direta representa uma função de sua entrada atual, ou seja, ela não tem nenhum estado interno além dos pesos propriamente ditos. Enquanto que em uma rede recorrente existem laços de realimentação, ou seja, as saídas da rede são usadas para alimentar suas próprias entradas. Isto significa que os níveis de ativação da rede formam um sistema dinâmico que pode exibir um estado estável, exibir oscilações ou até mesmo apresentar um comportamento caótico.

#### A. Implementação de Redes Neurais Artificiais no Ambiente FF

A solução aqui abordada envolve dois blocos funcionais: o aritmético e o caracterizador, mostrados nas Figuras 1 e 2 respectivamente. Eles devem ser configurados e interligados de maneira que o conjunto se comporte como um neurônio artificial, mostrado na Figura 3. No bloco aritmético, deve-se escolher o *Algorithm Type* como Somador Tradicional e preencher os ganhos das entradas de acordo com o treinamento realizado off-line.

No bloco caracterizador, escolhe-se até 21 pontos, como informado na seção anterior, para simular a função de ativação.

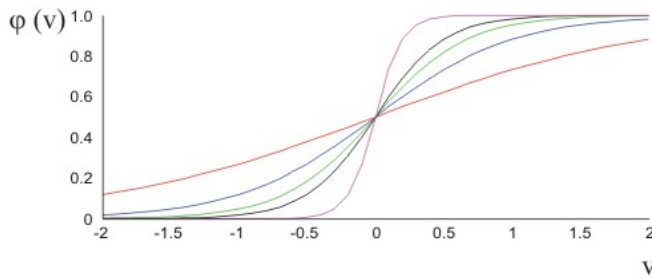


Fig. 4. Função escolhida para ser a função de ativação.

Neste trabalho escolheu-se a função sigmóide, mostrada na figura 4.

A escolha dos pontos deve ter como objetivo a minimização do erro entre a função sigmóide original e a função aproximada. Para essa escolha foi utilizada uma técnica de inteligência computacional conhecida como Algoritmos Genéticos. Para medir a qualidade da aproximação, a técnica utilizada se baseia numa função de avaliação, aqui definida como sendo o erro médio quadrático entre a função original e a função formada pela interligação dos pontos escolhidos. O conjunto dos pontos escolhidos resultantes, está mostrado na Figura 5.

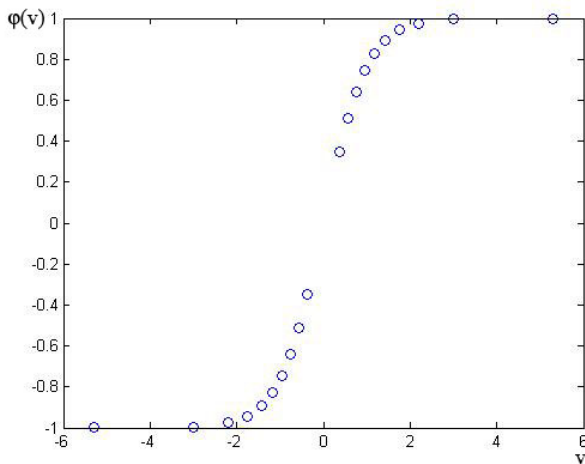


Fig. 5. Pontos escolhidos para formar a função de ativação.

Afim de se visualizar a qualidade da aproximação, sobrepôs-se as duas funções, como visto na Figura 6.

Ao se ligar a saída de um bloco funcional aritmético à entrada de um bloco funcional caracterizador de sinais, configurados como descrito acima, tem-se um neurônio artificial no ambiente FF. E é através da interligação desses neurônios que se constroem redes neurais, como mostrado na Sessão seguinte.

#### IV. AMBIENTE DE TESTES E RESULTADOS OBTIDOS

O ambiente de testes utilizado consiste de uma rede industrial *Foundation Fieldbus* com seis dispositivos de campo e uma *bridge*, que faz a interligação da rede industrial com os computadores na rede Ethernet, onde estão instalados os

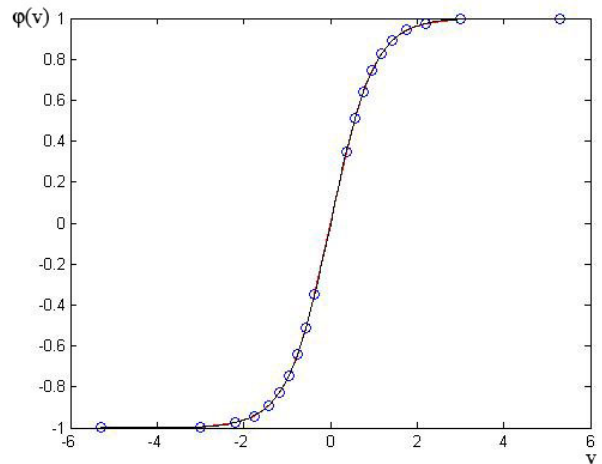


Fig. 6. As duas funções sobrepostas.

softwares de configuração. A Figura 7 representa a arquitetura da rede didática utilizada.

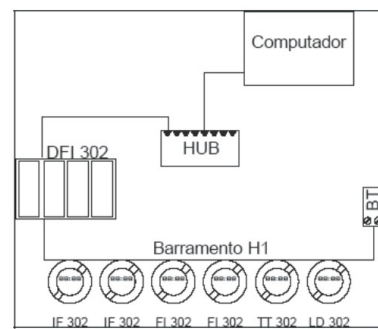


Fig. 7. Arquitetura da rede didática FF.

O procedimento citado anteriormente, pode ser implementado em qualquer dispositivo da rede FF, inclusive na *bridge*. A seguir, apresenta-se a implementação de dois exemplos.

##### A. Exemplo 1

Nesse primeiro exemplo treinou-se uma rede neural de arquitetura 1-3-1, como mostrado na Figura 8. para aprender o conjunto de treinamento entrada-saída ilustrados na figura 9.

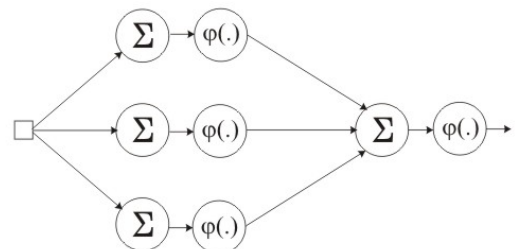


Fig. 8. Rede neural implementada para o exemplo 1.

O diagrama de blocos funcionais da Figura 10 implementa a rede neural da Figura 8 no ambiente FF.

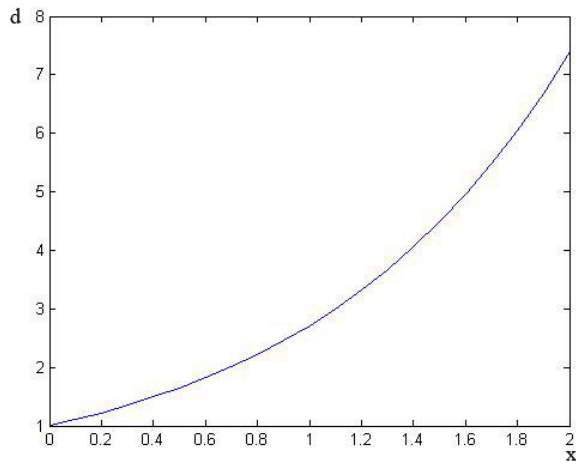


Fig. 9. Conjunto de dados de treinamento.

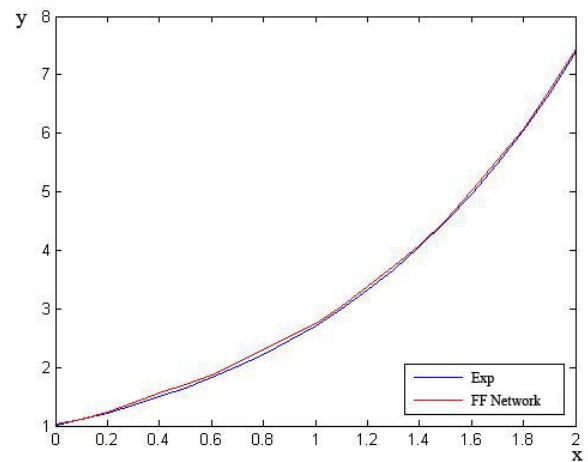


Fig. 11. Resposta da rede neural ao conjunto de dados de validação.

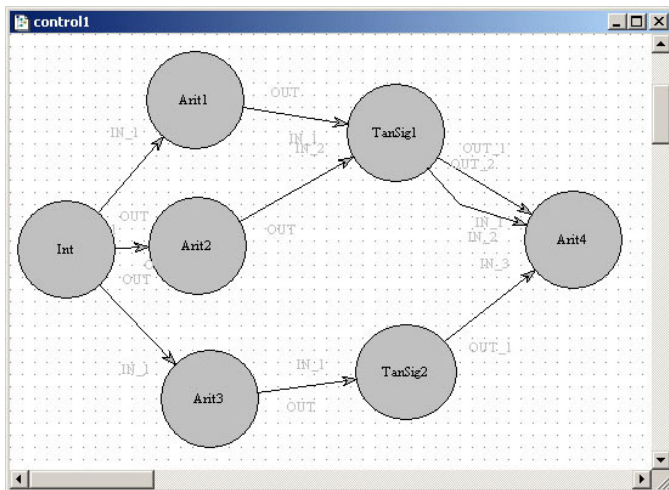


Fig. 10. Rede neural implementada no ambiente FF.

Para validação da implementação no ambiente FF, foi apresentado um conjunto de dados. Sua respectiva resposta, juntamente com o conjunto de dados originais podem ser conferidas na Figura 11.

O erro médio quadrático entre a resposta da rede no ambiente FF e o conjunto de dados original foi:

$$erro = 2,7 \times 10^{-3}$$

### B. Exemplo 2

O segundo exemplo, treina uma rede neural de arquitetura 2-3-1, mostrada na Figura 12 para aprender o conjunto de dados de três dimensões mostrado na Figura 13.

O erro médio quadrático alcançado nesse exemplo foi de:

$$erro = 2,8 \times 10^{-3}$$

O gráfico que mostra a validação para este exemplo está na Figura 14.

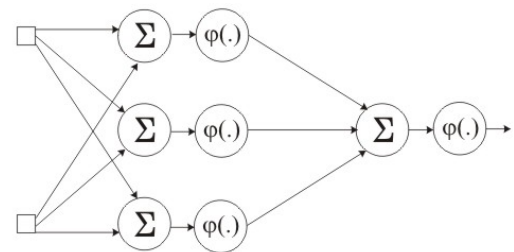


Fig. 12. Rede neural implementada para aprender o exemplo 2.

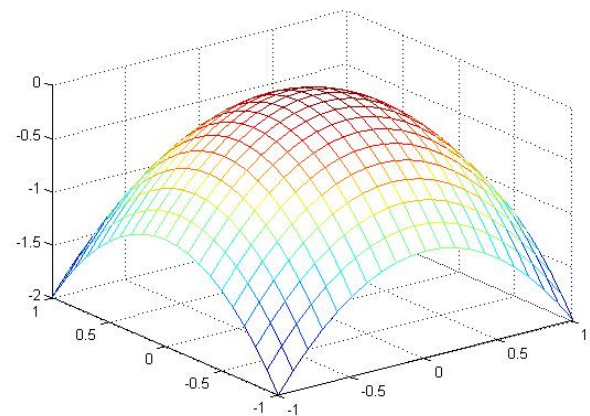


Fig. 13. Conjunto de treinamento 3D.

## V. CONCLUSÕES

Este trabalho aborda a importância da ferramenta matemática de redes neurais artificiais e cita algumas possíveis aplicações dela no ambiente industrial. Além disso, apresenta uma solução de implementação em ambientes de redes industriais *Foundation Fieldbus*, que utiliza apenas blocos funcionais padronizados. O que significa que a implementação é possível em qualquer, garantindo a interoperabilidade.

A validação, mostrada na sessão anterior permite avaliar o desempenho da rede e ajuda a decidir sobre a viabilidade

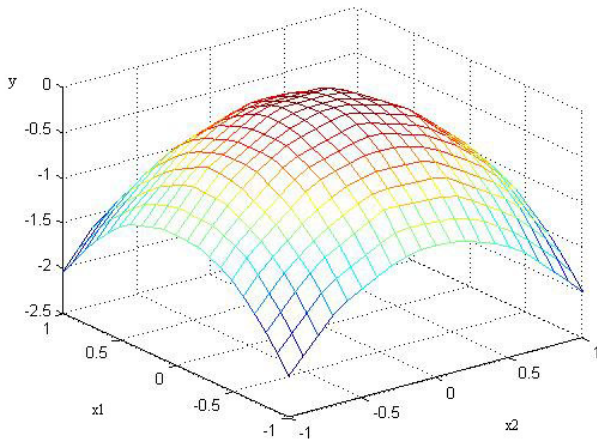


Fig. 14. Validação do exemplo 2.

de determinadas aplicações. As taxas de erros obtidas são consideradas aceitáveis em vários casos, o que comprova o êxito da solução aqui abordada.

Pretende-se em trabalhos futuros, aplicar as redes industriais FF em controle de processos não-lineares, fazendo uso da técnica apresentada neste texto.

#### REFERÊNCIAS

- [1] Berge, J., *Fieldbus for Process Control: Engineering, Operation and Maintenance*, ISA - The Instrumentation, System and Automation Society, 2001.
- [2] *Technical Information: Foundation Fieldbus*, Samson.
- [3] M. Nørgaard, O. Ravn, N. K. Poulsen and L. K. Hansen. Springer-Verlag, London Berlin Heidelberg, *Neural networks for modelling and control of dynamic systems* Springer-Verlag, 2000.
- [4] L. Fortuna; S. Graziani; M. G. Xibilia, *Soft sensors for product quality monitoring in debutanizer distillation columns*, Control Engineering Practice, 2004.
- [5] Yang, S. H.; Chen B. H.; Wang, X. Y., *Neural Network based fault diagnosis using unmeasurable inputs*, Engineering Application of Artificial Intelligence, 2000.
- [6] [www.fieldbus.org](http://www.fieldbus.org)
- [7] Haykin, Simon, *Redes Neurais: Princípios e Prática*, Bookman, 2001.
- [8] Bose, N. K.; Liang, P. *Neural Network Fundamentals with Graphs, Algorithms and Applications*, McGraw-Hill, 1996.
- [9] David E. Goldberg *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.