

# IMPLANTAÇÃO DE UM SENSOR VIRTUAL NO SISTEMA PIMS BASEADO EM REDES NEURAIS ARTIFICIAIS PARA FINS DE DIAGNÓSTICO DE PROCESSO\*

Cláudio Antunes de Oliveira<sup>1</sup>  
Fernando da Silva Dummer<sup>2</sup>  
João Marques Salomão<sup>3</sup>  
Hans Rolf Kulitz<sup>4</sup>

## Resumo

O objetivo deste trabalho é utilizar o vasto banco de dados de informações de processo armazenadas pelo InfoPlus.21, que é um exemplo de sistema *PIMS - Plant Information Management System*, para implementar um sensor virtual a partir de um modelo neural visando obter a tendência de resposta da variável consumo específico de uma caldeira. Propõe-se também, extrair diagnósticos do processo através do emprego da Lógica *Fuzzy*. As etapas de treinamento da rede neural e o desenvolvimento do diagnóstico envolveram simulações no *software Matlab*. Os resultados obtidos demonstraram-se bastante satisfatórios. Destaca-se a tendência de resposta do sensor virtual que apontou 97% de confiabilidade. Por fim entende-se que as propostas deste trabalho foram alcançadas, principalmente pela implementação inovadora do sensor virtual no *InfoPlus.21* por meio da ferramenta *SQLPlus*, a qual possibilitará novas aplicações no meio industrial.

**Palavras-chave:** PIMS; Rede neural; Lógica *fuzzy*; Diagnóstico de processo.

## IMPLEMENTATION OF VIRTUAL SENSOR ON SYSTEM PIMS BASED ON NEURAL NETWORKS FOR PROCESS DIAGNOSIS

### Abstract

The propose of this work is to use the process information stored on InfoPlus.21 database to develop a virtual sensor from a neural model in order to obtain the variable trend. It is also proposed to extract the process diagnosis using the Fuzzy Logic concepts. All simulations to train the neural network, extract the neural model and develop the fuzzy diagnosis, were developed using Matlab software. The main result obtained was 97% reliability in the response of the virtual sensor. Finally, it is understood that proposals of this study were achieved, mainly by innovative implementation of the virtual sensor on InfoPlus.21 through SQLPlus tool, which will allow new applications in similar industrial cases.

**Keywords:** PIMS; Neural network; Fuzzy logic; Process diagnosis.

<sup>1</sup> Engenheiro Eletricista, Especialização, Especialista em Automação, Gerência de Informática e Automação de Processo, ArcelorMittal Tubarão, Serra, ES e Brasil.

<sup>2</sup> Engenheiro Eletricista, Especialista em Automação, Gerência de Informática e Automação de Processo, ArcelorMittal Tubarão, Serra, ES e Brasil.

<sup>3</sup> Doutor em Engenharia Elétrica, Professor titular, Departamento de Engenharia Elétrica, Instituto Federal do Espírito Santo, Vitória, ES e Brasil.

<sup>4</sup> Doutor em Engenharia Elétrica, Professor titular, Departamento de Engenharia Elétrica, Instituto Federal do Espírito Santo, Vitória, ES e Brasil.

## 1 INTRODUÇÃO

Os sistemas PIMS – *Process Information Management Systems* são sistemas que adquirem dados de processo de diversas fontes, os armazenam num banco de dados históricos e os disponibilizam através de diversas formas de representação [1]. Uma destas formas é mostrada na figura 1 onde é possível analisar tendências históricas para investigação de falhas. A partir do banco de dados histórico é possível empregar técnicas de modelagem visando extrair conhecimento sobre o comportamento do processo ou de uma variável de interesse. Dentre as técnicas de modelagem empregadas, podemos destacar a obtenção de modelos a partir das redes neurais artificiais.

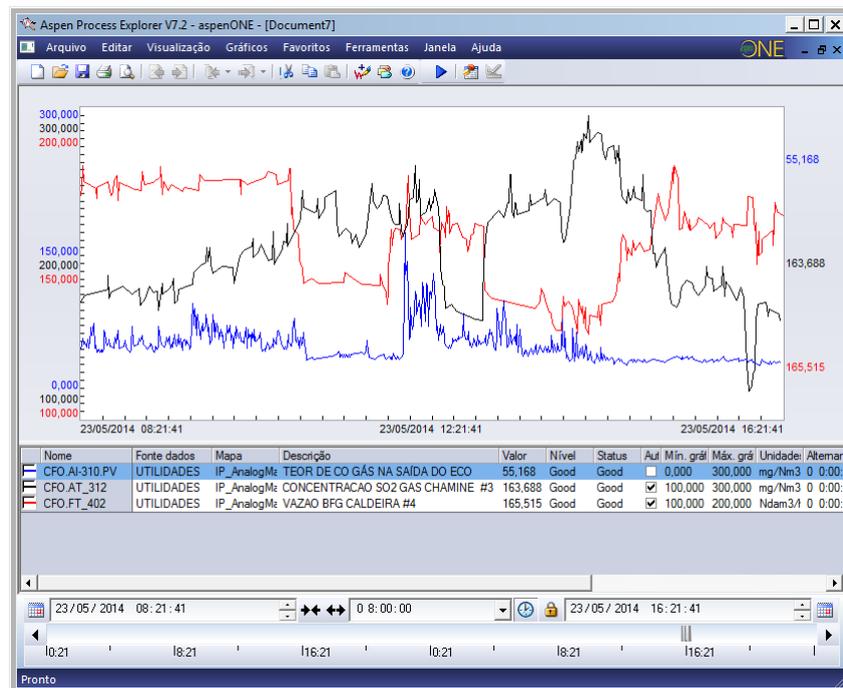
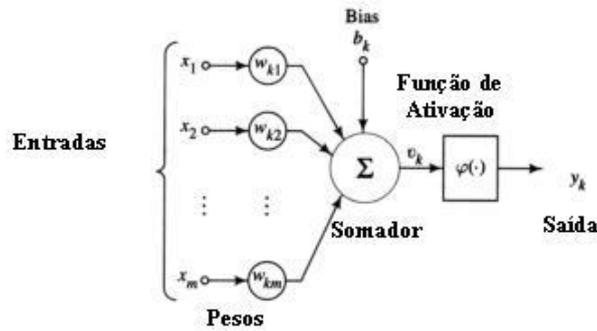


Figura 1 – Gráfico do InfoPlus.21

Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. A operação de uma unidade de processamento, proposta por McCulloch e Pitts em 1943, pode ser resumida da seguinte maneira:

- Sinais são apresentados à entrada;
- Cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade;
- É feita a soma ponderada dos sinais que produz um nível de atividade;
- Se este nível de atividade exceder um certo limite (threshold) a unidade produz uma determinada resposta de saída [2].

Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural. O diagrama da figura 2 mostra o modelo de um neurônio.



**Figura 2 – Modelo de neurônio**

Em termos matemáticos, podemos descrever um neurônio k escrevendo o seguinte par de equações:

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

e

$$y_k = \varphi(u_k + b_k)$$

Onde  $x_1, x_2, \dots, x_m$  são os sinais de entrada;  $w_{k1}, w_{k2}, \dots, w_{km}$  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;  $\varphi(.)$  é a função de ativação; e  $y_k$  é o sinal de saída do neurônio [3].

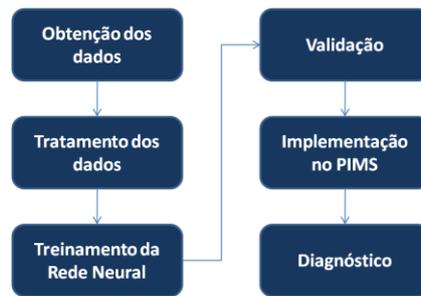
Adotando-se esta técnica computacional, será abordado neste trabalho, o desenvolvimento de um sensor virtual bem com a sua implementação no *InfoPlus.21*, utilizando-se a ferramenta *SQLPlus*. O sensor virtual pode ser entendido como sendo uma medida não física, inferida em tempo real por meio de um modelo que descreva o comportamento do processo ou da variável física de interesse. Este modelo geralmente é obtido através de medidas físicas do processo que apresentam correlação com a variável que se deseja inferir e podem ser obtidos por diferentes tipos de técnicas. Há várias formas de se classificar técnicas de modelagem. Uma delas agrupa os métodos em três categorias denominadas modelagem caixa branca, modelagem caixa preta e modelagem caixa cinza [4].

O texto está organizado em 4 seções. A seção 2 descreve os materiais e métodos enquanto que os resultados são abordados na seção 3. Para finalizar, a seção 4 apresenta as conclusões e trabalhos futuros.

## 2 MATERIAIS E MÉTODOS

Este trabalho foi desenvolvido de acordo com as 6 etapas mostradas na figura 3, as quais serão detalhadas nos sub-tópicos seguintes. O software Microsoft Excel 2007 foi utilizado para coleta e tratamento dos dados e o software Matlab R2013a foi empregado para realização das simulações envolvendo a tanto a Rede Neural quanto a Lógica *Fuzzy*.

\* Contribuição técnica ao 18º Seminário de Automação e TI Industrial, 23 a 26 de setembro de 2014, São Paulo, SP, Brasil.



**Figura 3** – Etapas do trabalho

## 2.1 Obtenção dos Dados

A massa de dados necessária para elaboração deste trabalho foi obtida de duas fontes distintas. A maior parte dela e abrangendo as variáveis de processo de entrada do modelo neural, foi obtida diretamente do InfoPlus.21, utilizando-se o recurso de exportar dados para o Microsoft Excel. Este recurso pode ser adicionado por meio da opção Suplementos do próprio Excel. Para complementar a massa de dados de treinamento, a variável de interesse do modelo neural foi extraída do Sistema de Otimização de Processo de Nível 2, onde esta variável é calculada. Assim, as 8 variáveis de processo mostradas na tabela 1, foram amostradas a uma frequência de 60s por um período de 8 meses, totalizando uma massa de dados composta por 345.600 minutos.

**Tabela 1** – Variáveis de entrada e saída

Variável	Descrição	Unidade	Tipo
1	Vazão de BFG – Blast Furnace Gas	Nm <sup>3</sup> /h	Entrada
2	Vazão de COG – Coke Oven Gas	Nm <sup>3</sup> /h	Entrada
3	Vazão de Alcatrão	Nm <sup>3</sup> /h	Entrada
4	Vazão de Vapor	Nm <sup>3</sup> /h	Entrada
5	Poder calorífico inferior do COG	Kcal/m <sup>3</sup>	Entrada
6	Poder calorífico inferior do BFG	Kcal/m <sup>3</sup>	Entrada
7	Poder calorífico inferior do Alcatrão	Kcal/m <sup>3</sup>	Entrada
8	Consumo específico da caldeira	-	Saída

## 2.2 Tratamento dos Dados

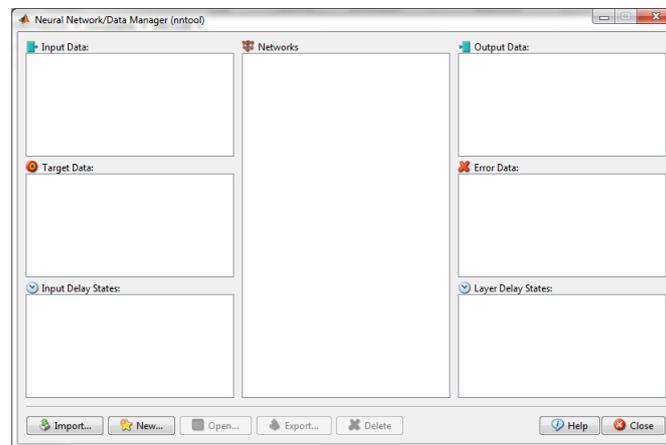
Para garantir um adequado processo de aprendizagem e/ou de treinamento da rede neural, o qual será abordado na próxima seção, foi necessário analisar a massa de dados coletada com o intuito de identificar condições atípicas de processo ou falhas nas medições que poderiam resultar em uma generalização incorreta da rede neural. Assim nesta etapa foram excluídos da massa de dados:

- Períodos de parada programada da planta;
- Problema de medição devido falha e/ou calibração de instrumento;
- Períodos em que a variável de interesse apresentou erro de cálculo no Sistema de Otimização de Processo de Nível 2.

## 2.3 Treinamento da Rede Neural (Processo de Aprendizagem)

Nesta etapa será abordado o treinamento da rede neural empregada neste trabalho, que é do tipo *perceptron* multicamadas. O algoritmo de treinamento utilizado foi o *backpropagation*. O objetivo desta etapa é a obtenção do modelo neural utilizando a

ferramenta *NNtool*, a qual está disponível no software *Matlab*. A figura 4 mostra a tela principal desta ferramenta. Para acessá-la, basta digitar na linha de comando do *Workspace* do *Matlab* a palavra *nntool* e digitar *Enter*.



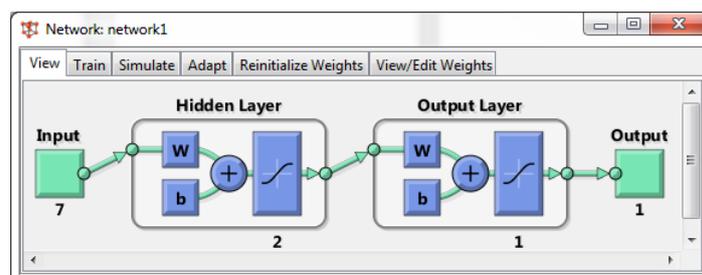
**Figura 4** – Tela principal da ferramenta *NNTool*

De forma geral, necessitamos basicamente de duas operações para realizar o treinamento da rede neural através da ferramenta *NNtool*:

- Operação 1: através do botão *Import* efetuamos o carregamento das variáveis de entrada por meio do campo *Input Data* e posteriormente da variável de saída por meio do campo *Target Data*. Os dados de entrada e saída devem ser carregados primeiramente no *Workspace* do *Matlab* na forma de vetor transposto antes de iniciar a operação 1;
- Operação 2: uma vez carregados os dados de entrada e saída, defini-se os parâmetros da rede neural através do botão *New*. Os parâmetros configurados de forma mais usual são: o tipo de rede, o algoritmo de treinamento, o número de neurônios na camada escondida e o número de neurônios na camada de saída.

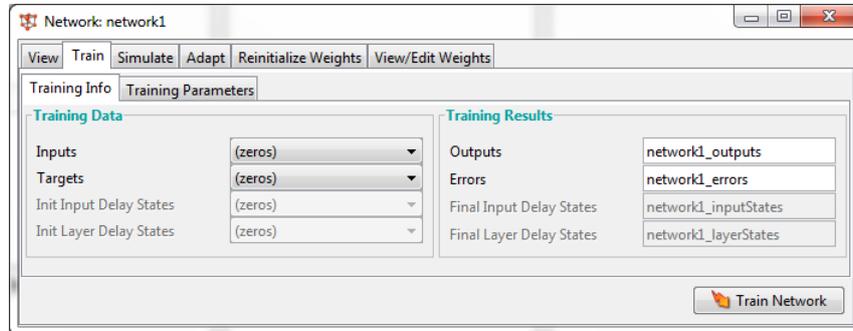
Após a configuração da rede neural é possível visualizar sua arquitetura por meio do botão *Open* a partir da tela principal da ferramenta *NNtool*. Uma nova tela será aberta e na guia *View* é possível visualizar a referida arquitetura. Na figura 5 é mostrada a arquitetura da rede neural utilizada na etapa de treinamento deste trabalho, a qual possui 4 blocos, descritos conforme segue:

- *Input* - um vetor contendo 7 variáveis de entrada;
- *Hidden Layer* – uma camada escondida composta por dois neurônios;
- *Output Layer* – uma camada de saída composta por um neurônio;
- *Output* – um vetor contendo 1 variável de saída.



**Figura 5** – Arquitetura da rede neural utilizada

Uma vez que a arquitetura da rede neural está correta de acordo com a configuração realizada por meio das operações 1 e 2 citadas anteriormente, o treinamento propriamente dito da rede neural é realizado através da guia *Train*, conforme ilustrado na figura 6. Nesta guia, basta informar novamente os vetores de entrada e saída nos campos *Inputs* e *Targets* respectivamente e clicar no botão *Train Network*.



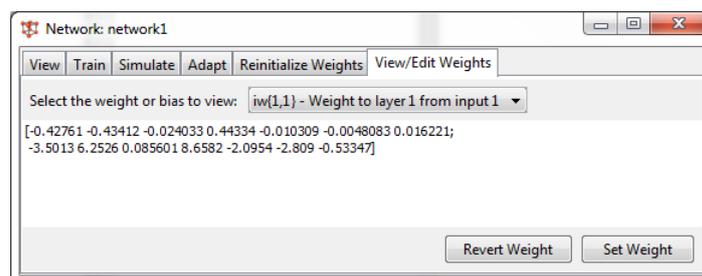
**Figura 6** – Tela de treinamento da rede neural

Realizado o treinamento e este apresentado resultado satisfatório de acordo com o *erro quadrático médio* estipulado, podemos obter os parâmetros do modelo, os quais são descritos na tabela 2. Estes parâmetros são acessados através da guia *View/Edit Weights*, conforme figura 7.

**Tabela 2** – Parâmetros do modelo neural

<i>Parâmetro</i>	<i>Descrição</i>
$iw\{1,1\}$ – Weight to layer 1 from input 1	Pesos do neurônio 1
$iw\{2,1\}$ – Weight to layer	Pesos do neurônio 2
$b\{1\}$ – Bias to layer 1	Bias do neurônio 1
$b\{2\}$ – Bias to layer 2	Bias do neurônio 2

Estes parâmetros são utilizados no desenvolvimento do algoritmo que implementa o sensor virtual no InfoPlus.21. Este desenvolvimento será abordado no subitem seguinte.



**Figura 7** – Pesos da camada de entrada

## 2.4 Implementação do Sensor Virtual no InfoPlus.21

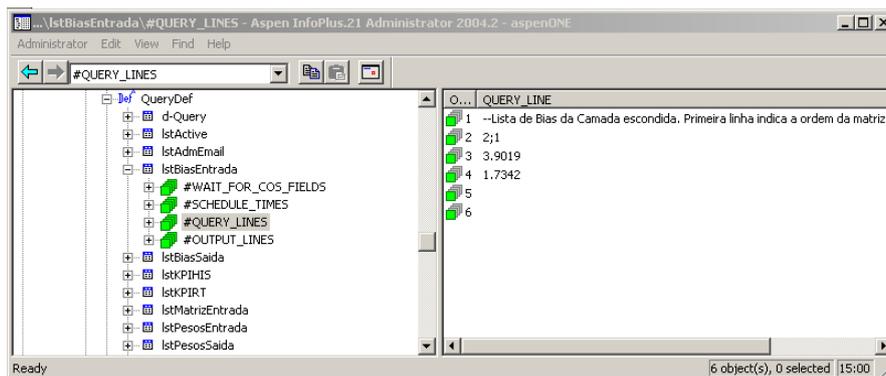
Utilizando o recurso da ferramenta *SQLPLUS* disponível no pacote de instalação do InfoPlus.21, foi desenvolvido um algoritmo em linguagem *SQL* para implementar o sensor virtual. O algoritmo executa as 8 etapas mostradas na figura 8.

\* *Contribuição técnica ao 18º Seminário de Automação e TI Industrial, 23 a 26 de setembro de 2014, São Paulo, SP, Brasil.*

1. Carrega as variáveis de entrada.
2. Normaliza as variáveis de entrada.
3. Multiplica cada entrada normalizada do neurônio pelo seu respectivo peso.
4. Soma os resultados da multiplicação de cada entrada do neurônio.
5. Adiciona o bias de cada neurônio ao resultado da soma anterior obtendo a saída do neurônio.
6. Submete o valor de saída do neurônio à função de ativação.
7. Obtém a saída da rede neural e normaliza a variável de saída.
8. Grava o resultado obtido na base de dados do PIMS.

**Figura 8** – Etapas de execução do algoritmo

A partir dos parâmetros de peso e bias obtidos após o treinamento da rede neural e conforme mostrado na tabela 2 do sub-item 2.3, foi implementado inicialmente no *InfoPlus.21*, a parametrização de 5 *Queries*, utilizando a opção *QueryDef*. As *Queries* criadas foram: *IstBiasEntrada*, *IstBiasSaida*, *IstPesosEntrada*, *IstPesosSaida* e *IstMatrizEntrada*. Um exemplo desta parametrização é mostrado na figura 9, onde podemos visualizar o arquivo *IstBiasEntrada* e o conteúdo da sua configuração efetuada a partir da opção *QUERY\_LINES*. Nesta figura, os valores 3,9019 e 1,7342 correspondentes as linhas 3 e 4 respectivamente, são os parâmetros de *bias* de cada um dos dois neurônios da camada escondida.



**Figura 9** – Configuração da matriz *IstBiasEntrada*

Posteriormente foi elaborado o algoritmo do sensor virtual a partir da ferramenta *SQLPlus*. No início da sua execução, o algoritmo carrega cada uma das 5 *Queries*, pois estas contém os parâmetros do modelo neural obtido na etapa de treinamento. Um fragmento do algoritmo bem como o resultado de uma simulação executada a partir da ferramenta *SQLPlus* é mostrada na figura 10 e na figura 11 é mostrada um histórico da tendência de resposta do sensor virtual implantado no *InfoPlus.21*.

\* Contribuição técnica ao 18º Seminário de Automação e TI Industrial, 23 a 26 de setembro de 2014, São Paulo, SP, Brasil.



```

Aspen SQLplus 2004.2 - aspenONE - UTILIDADES - [PC:RedeNeural]
File Edit Record View Debug Query Window Help
--CALCULO DA PROD_PESO_ENTRADA
FOR i = 1 TO UBound(PESO,1) DO
  FOR j = 1 TO UBound(ENTRADA,2) DO
    FOR k = 1 TO UBound(ENTRADA,1) DO
      PROD_PESO_ENTRADA[i,j] = PROD_PESO_ENTRADA[i,j] + (PESO[i,k] * ENTRADA[k,j]);
    END; --FOR DO K
  END; --FOR DO J
END; --FOR DO I
exibeMatriz('PRODUTO_PESO_ENTRADA', PROD_PESO_ENTRADA);
--PREENCHE MATRIZ BIAS
FOR
  SELECT TRIM(QUERY_LINE) AS VALOR, OCCNUM AS OCCORR FROM 1stBiasEntrada WHERE OCCNUM > 2 AND TRIM(QUERY_LINE) <> ''
  order by OCCNUM
DO
  BIAS[OCCORR - 2,1] = VALOR;
END;
exibeMatriz('BIAS_ENTRADA',BIAS);

PRODUTO PESO_SAIDA CAHADADESCONHIDA [1,1] = 1131.92639980688
*****
BIAS_SAIDA[1,1] = -1132.567
RESULTADO FINAL (Ates de aplicar ativação) = -0.640250583744319
RESULTADO FINAL (Ates de aplicar ativação) = -0.640250583744319
RESULTADO FINAL (Após aplicar ativação) = -0.565070148282667
RESULTADO FINAL (Normalizado) = 515.419269768864
RESULTADOFINAL = 515.419269768864
Ready
  
```

Figura 10 – Exemplo de simulação na ferramenta *SQLPlus*

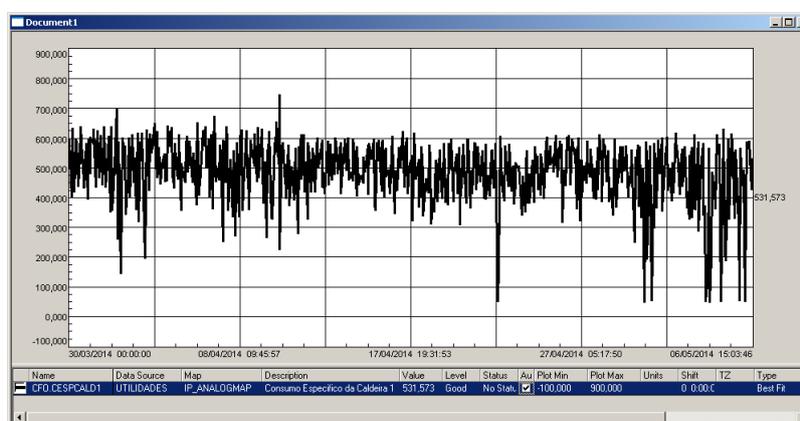


Figura 11 – Tendência de resposta do sensor virtual

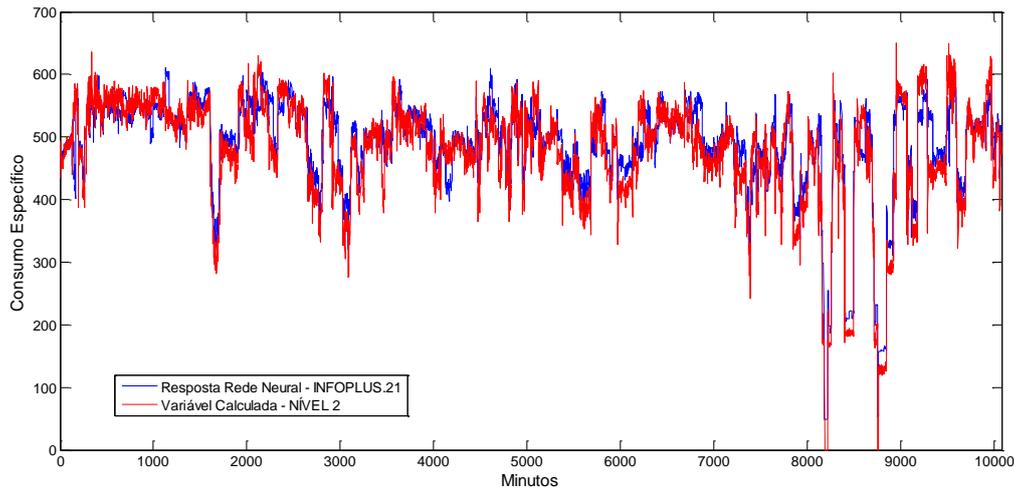
## 2.5 Validação do Algoritmo

Nesta etapa de validação do algoritmo, comparou-se a resposta gerada pelo *InfoPlus.21* com a resposta gerada pelo *Matlab*. Para isso, a rede neural obtida através da ferramenta *NNTool*, foi exportada para o *Workspace* do *Matlab* para permitir a obtenção da resposta deste sistema. Assim, por meio da função *sim* e de um conjunto aleatório de variáveis de entrada obtido diretamente do *InfoPlus.21*, determinou-se a resposta do *Matlab*. A tabela 3 mostra a comparação entre estas duas respostas.

Tabela 3 – Comparação entre as resposta do *Matlab* e do *InfoPlus.21*

Amostra	Resposta <i>Matlab</i>	Resposta <i>InfoPlus.21</i>	Erro
1	594,06	594,08	0,02
2	599,99	600,01	0,02
3	565,88	565,91	0,03

Para complementar esta etapa de validação, comparou-se também a tendência de resposta do algoritmo implementado no *InfoPlus.21* com a tendência da variável calculada no Sistema de Otimização de Processo de Nível 2. A figura 12 mostra a comparação destas duas respostas num período de 1 semana.



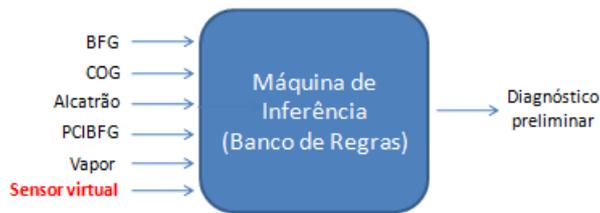
**Figura 12** – Resposta do *InfoPlus.21* e do Nível 2

## 2.6 Diagnóstico

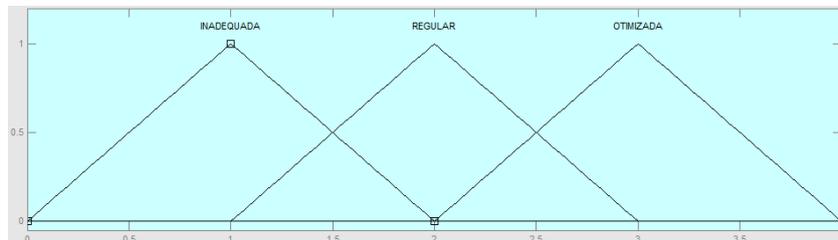
O processo industrial escolhido para desenvolvimento deste sensor virtual foi uma caldeira, a qual faz parte de uma Central Termoelétrica. Neste tipo de processo, um dos principais indicadores é o *consumo específico da caldeira*, pois através dele, o operador determina suas ações de controle, visando maximizar a geração de vapor com menor consumo de combustível. Diante disto, propõe-se a utilização de outra técnica computacional - a *Lógica Fuzzy*, para inferir o diagnóstico deste processo.

A *Lógica Fuzzy* é uma técnica que incorpora a forma humana de pensar em um sistema de controle. Um controlador *fuzzy* típico pode ser projetado para comportar-se conforme o raciocínio dedutivo, isto é, o processo que as pessoas utilizam para inferir conclusões baseadas em informações que elas já conhecem. Por exemplo, operadores humanos podem controlar processos industriais e plantas com características não-lineares e até com comportamento dinâmico pouco conhecido, através de experiência e inferência de relações entre as variáveis do processo. A *Lógica Fuzzy* pode capturar este conhecimento em um controlador *fuzzy*, possibilitando a implementação de um controlador computacional com desempenho equivalente ao do operador humano [5].

Adotando-se esta característica de inferência da *Lógica Fuzzy*, foi desenvolvida no Matlab uma proposta de diagnóstico do processo de combustão da caldeira, conforme ilustrado na figura 13. Como podemos perceber, as variáveis de entrada da máquina de inferência são as principais variáveis do processo de combustão da caldeira, acrescida da resposta do sensor virtual – o consumo específico da caldeira. A partir dessas entradas, a máquina de inferência, composta por um banco de regras que mapeia as condições de processo, ou seja, possui o conhecimento prévio do processo, gera o diagnóstico. A classificação da combustão por sua vez é inferida como: inadequada, regular e otimizada conforme mostrada na figura 14.



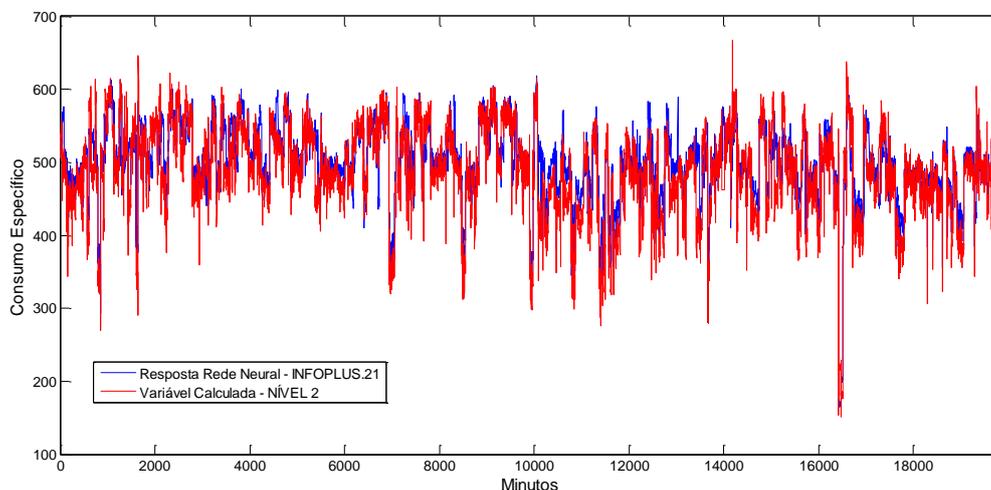
**Figura 13** – Máquina de inferência *fuzzy*



**Figura 14** – Classificação do diagnóstico da combustão

### 3 RESULTADOS E DISCUSSÃO

Na figura 15 é mostrada a tendência de resposta do sensor virtual por um período de 30 dias. Como o interesse nesta aplicação é a tendência da variável *consumo específico*, inferida por meio do modelo neural, entende-se que o resultado alcançado foi bastante expressivo, o qual registrou 97% de confiabilidade na indicação do sensor virtual implementado no *InfoPlus.21*. Neste mesmo período, foi calculado o erro médio percentual entre a resposta do sensor virtual e o cálculo da variável de interesse realizado pelo Nível 2, o qual é apresentado na tabela 4.



**Figura 15** – Tendência de resposta do sensor virtual

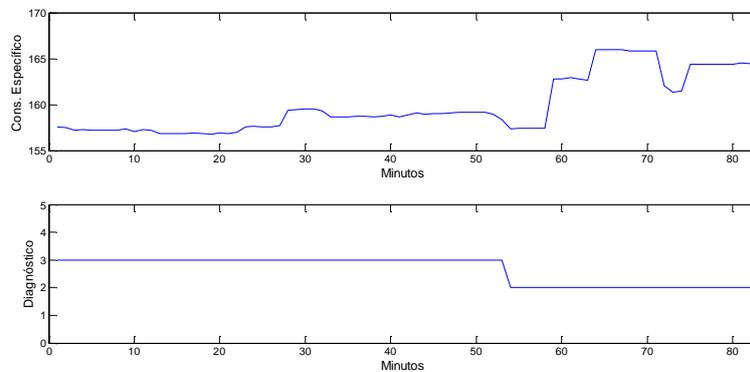
Neste período, o erro médio percentual máximo foi verificado na semana 4, atingindo -3,42%. Vale ressaltar que este erro têm apresentado sinal negativo, apontando que em relação a variável calculada no Nível 2, a tendência de resposta do sensor virtual está menor. Levando-se em consideração a arquitetura da rede neural utilizada no treinamento, a qual utilizou apenas 2 neurônios na camada escondida, conforme abordado no subitem 2.3, sugere-se aumentar o número de neurônio desta camada, refazer a etapa de treinamento e avaliar o erro médio percentual. Porém, como o propósito é obter uma tendência da variável de interesse, optou-se por trabalhar com

uma arquitetura mais enxuta, visando menor esforço computacional na implementação do sensor virtual no *InfoPlus.21*.

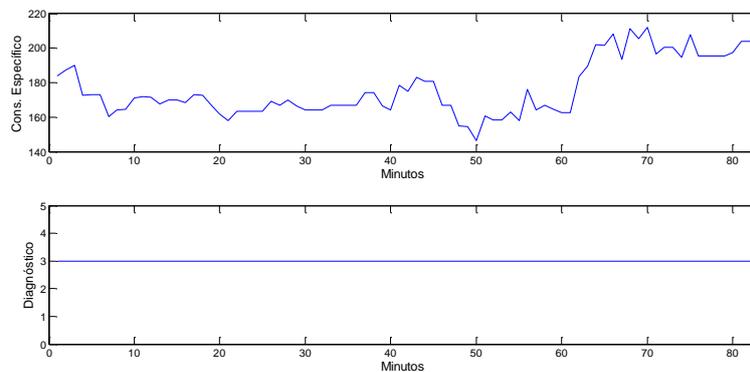
**Tabela 4 – Erro médio apresentado pela rede neural**

<i>Período</i>	<i>Consumo Específico (sensor virtual)</i>	<i>Consumo Específico (calculado)</i>	<i>Erro Médio (%)</i>
Semana 1	506,65	517,56	-2,11
Semana 2	465,91	482,27	-3,39
Semana 3	478,29	490,40	-2,47
Semana 4	423,01	438,01	-3,42

Em relação ao diagnóstico proposto, as figuras 16 e 17 demonstram exemplos de resposta da máquina de inferência. A partir de condições reais de processo, o diagnóstico apresentado foi uma transição de otimizado para regular (o processo de combustão está com tendência de piorar) e otimizado, respectivamente.



**Figura 16 – Diagnóstico preliminar de transição**



**Figura 17 – Diagnóstico preliminar de combustão otimizada**

## 4 CONCLUSÃO

Foi possível demonstrar neste trabalho, que as duas técnicas computacionais empregadas, demonstraram-se bastante aplicáveis em problemas de modelagem, onde a extração de conhecimento a partir de dados históricos de processo é de fundamental importância para a tomada de decisão do ponto de vista operacional.

Em relação à tendência de resposta do sensor virtual implantado no *InfoPlus.21*, considera-se que o resultado obtido de 97% de confiabilidade foi bastante satisfatório para a aplicação abordada neste trabalho, porém entende-se que

\* Contribuição técnica ao 18º Seminário de Automação e TI Industrial, 23 a 26 de setembro de 2014, São Paulo, SP, Brasil.

melhorias na obtenção do modelo neural podem ser realizadas objetivando aumentar a confiabilidade do sensor virtual.

O diagnóstico proposto, embora tenha abordado apenas os combustíveis utilizados na caldeira, também apresentou bons resultados nas simulações. Porém como objeto de trabalho futuro, estuda-se a inclusão de novas regras na máquina de inferência, permitindo com isso enriquecer o diagnóstico proposto com outras condições de processo, além da implementação deste diagnóstico no *InfoPlus.21* ou no Sistema de Otimização de Processo de Nível 2.

Por fim, entende-se que o algoritmo desenvolvido no *InfoPlus.21* permitirá alavancar novas aplicações de sensores virtuais. Dentre estas aplicações, pretende-se utilizar o banco de dados históricos do *InfoPlus.21*, para treinar uma rede neural preditiva visando obter antecipadamente o comportamento do processo.

## **Agradecimentos**

Agradeço ao corpo docente do curso de Pós Graduação em Sistemas Inteligentes Aplicados a Automação do Instituto Federal do Espírito Santo, pelo conhecimento proporcionado, o qual possibilitou a base teórica necessária para desenvolver este trabalho. Agradeço também aos colegas da ArcelorMittal Tubarão do setor de Automação e das empresas parceiras que apoiaram este trabalho.

## **REFERÊNCIAS**

- 1 PIMS Plant Information Management System, Uma introdução. [acesso em 19 mar. 2014] 1-22. Disponível em:  
[http://www.das.ufsc.br/~rabelo/Ensino/DAS5316/MaterialDAS5316/Sistemas%22MES%20MES%20&%20PIMS/PIMS MES systems.pdf](http://www.das.ufsc.br/~rabelo/Ensino/DAS5316/MaterialDAS5316/Sistemas%22MES%20MES%20&%20PIMS/PIMS%20MES%20systems.pdf)
- 2 Redes neurais artificiais. [acesso em 11 abr. 2014] Disponível em:  
<http://www.icmg.usp.br/pessoas/andre/research/neural/>
- 3 Haykin S. Redes Neurais: princípios e prática. 2.ed. Porto Alegre: Bookman; 2001.
- 4 Aguirre LA. Introdução à Identificação de Sistemas: técnicas lineares e não lineares aplicadas a sistemas reais. 3.ed. rev. e ampl. Belo Horizonte: Editora UFMG; 2007.
- 5 Simões MG, Shaw IS. Controle e Modelagem Fuzzy. 2.ed. rev. e ampl. São Paulo: Blucher; 2007.

## **BIBLIOGRAFIA**

- 1 Neural Network Toolbox, User's Guide. [acesso em 11 fev. 2014] Disponível em:  
[http://www.mathworks.com/help/pdf\\_doc/nnet\\_ug.pdf](http://www.mathworks.com/help/pdf_doc/nnet_ug.pdf)
- 2 Fuzzy Logic Toolbox. [acesso em 11 fev. 2014] Disponível em:  
<http://www.mathworks.com/products/fuzzy-logic/>