Revista EIA, ISSN 1794-1237 / Year XI / Volume 11 / Issue N.22 / July-December 2014 / pp. 35-44
Technical-scientific biannual publication / Escuela de Ingeniería de Antioquia —EIA—, Envigado (Colombia)

PARAMETER CALIBRATION FOR ELECTRIC ARC FURNACE MODELS USING SIMULATION AND NEURAL NETWORKS

Mauricio Alexánder Álvarez López
Carlos Alberto Baena
Jesser James Marulanda

ABSTRACT

Electric arc furnaces provide a relatively simple way for melting metals. They are used in the production of highly purified steel, aluminium, copper and other metals. However, they are considered the most damaging load for electrical power systems. It is very important, therefore, to have arc furnace models that can determine the behavior of this type of load with a high degree of accuracy. In this way, it would be possible to assess the impact in terms of power quality indices for the power system to which they are connected. When using electric arc furnace models in practice, a key issue is the calibration of the model's parameters. In this paper, we show a procedure for calibrating all the parameters of an AC electric arc furnace model using real measurements of voltages and currents. A multilayer neural network is used as an emulator of the electric arc furnace model. The neural network is trained using data obtained from the simulation of the electric arc furnace model implemented in Matlab®-Simulink®. Once the network is trained, the parameters of interest are obtained by solving an inverse problem. The results obtained show a maximum percentage error of 4.1% for the rms value of the current involved in the electrical arc.

KEY WORDS: Electric arc furnace, calibration of parameters, neural networks, Latin Hypercube, computer emulation.

CALIBRACIÓN DE LOS PARÁMETROS DE UN MODELO DE HORNO DE ARCO ELÉCTRICO EMPLEANDO SIMULACIÓN Y REDES NEURONALES

RESUMEN

El horno de arco eléctrico proporciona un medio relativamente simple para la fusión de metales. Se utiliza en la producción de acero de alta pureza, aluminio, cobre, plomo, entre otros metales. Sin embargo, los hornos de arco son considerados como la carga más nociva para el sistema eléctrico de potencia. Por consiguiente, resulta de gran importancia contar con modelos de horno de arco que permitan determinar con alto grado de aproximación el comportamiento de este tipo de carga, puesto que se podría evaluar su impacto en términos de índices de calidad de energía para el sistema de potencia al cual se conecten. Uno de los principales problemas que surge al utilizar los modelos matemáticos de arco eléctrico consiste en la calibración de los parámetros que describen la dinámica del modelo. En este documento se muestra un procedimiento para calibrar todos los parámetros de un modelo de horno de arco eléctrico de corriente alterna, dadas

[1] Ph,D. Computer Science. Associate professor, Universidad Tecnológica de Pereira.
[2] Universidad Tecnológica de Pereira. Pereira, Colombia.
[3] Electrical engineer. Professor of Universidad Tecnológica de Pereira. Pereira, Colombia.

Correspondence author: Álvarez-López, M.A. (Mauricio Alexánder). Universidad Tecnológica de Pereira: Carrera 27 #10-02, Pereira (Colombia). Tel: (576) 3137300.
Email: malvarez@utp.edu.co

Paper history:
Paper received on: 8-IV-2013 / Approved: 27-III-2014
Available online: December 30 2014
Open discussion until December 2015

DOI: http:/dx.doi.org/10.14508/reia.2014.11.22.39-50

mediciones reales de tensiones y corrientes. Se utiliza una red neuronal multicapa como emulador del modelo del horno. La red neuronal se entrena empleando datos de simulación obtenidos del modelo del horno implementado en el entorno Matlab®-Simulink®. Una vez entrenada la red, los parámetros de interés se obtienen resolviendo un problema inverso. Los resultados obtenidos muestran un error máximo de 4.1 % en el valor eficaz de las corrientes del arco eléctrico.

PALABRAS CLAVES: horno de arco; calibración de parámetros; redes neuronales; *Latin Hypercube*, emulación por computador.

CALIBRAÇÃO DE PARÂMETROS DE UM MODELO USANDO FORNO DE ARCO ELÉTRICO EMPREGANDO SIMULAÇÃO E REDES NEURAIS

RESUMO

O forno a arco elétrico fornece um meio relativamente simples para a fusão de metais. Ele é usado na produção de ferro de alta pureza, alumínio, cobre, chumbo, e outros metais. No entanto, os fornos a arco são considerados a carga mais prejudicial no sistema elétrico de potência. Por conseguinte, é muito importante dispor de modelos de forno de arco para determinar com um elevado grau de aproximação o comportamento deste tipo de carga, uma vez que poderia avaliar o seu impacto em termos de índices de qualidade de energia para o sistema potencial para o que eles se conectam. Um dos principais problemas que surgem quando se utiliza modelos matemáticos do arco elétrico é a calibração dos parâmetros que descrevem dinâmica do modelo. Este documento apresenta um método para calibrar todos os parâmetros de um modelo de forno de arco elétrico de tensao alterna, dadas as medidas reais de tensões e correntes. Utiliza-se uma rede neural de multicamadas com um emulador de modelo de forno. A rede neural é treinada usando dados de simulação obtidos do modelo de forno usado no ambiente Matlab®-Simulink®. Uma vez seja treinada a rede, os parâmetros de interesse são obtidos através da resolução de um problema inverso. Os resultados obtidos mostram um erro máximo de 4,1 % no valor eficaz correntes de arco elétrico.

PALAVRAS-CHAVE: Fornos de arco; Calibração de parâmetros; Redes neurais; Latin Hypercube; Emulação de computador.

1. INTRODUCTION

The increase in electrical installations with electric arc furnaces among their loads has become of great interest to energy companies given that this load is considered the most damaging for the electrical power system in terms of power quality.

In general, the functions of an electric arc furnace are divided between the stages of fusion and refining. In the fusion stage, pieces of materials to be fused continually short circuit the furnace's electrodes, causing variations in the equivalent impedance of the electrodes' electrical circuit and, therefore, random fluctuations in the circuit's currents. These current fluctuations lead to variations in reactive power and momentary losses of voltage (flickers) in the load's connection busbar and in other nearby busbars. In the refining stage, the circuit's impedance variations decrease, lessening the impact on the power system. Electric arc furnaces are also known for being a source of harmonics, creating undesirable operating conditions for elements connected to the electrical network.

Therefore, being able to model the behavior of an electric arc furnace has become very important for energy companies (among others) in that it would give them a computational tool with which to know the impact a furnace could have on the energy system or design compensation systems like the static synchronous compensator (D-StatCom) or the static reactive power compensator (SVC) (García Cerrada et al., 2000).

However, one of the large problems that arise when using one of these electric arc furnace models in practice is calibrating their parameters. The related literature includes articles in which the parameters are heuristically tuned based on real measurements of the short-term flicker severity index (Pst) or based on the furnace's nominal power. Collantes & Gómez (1997) present a methodology for estimating parameters with real voltage measurements using the toolbox System Identification de Matlab®. Alves et al. (2010) adjust the parameters to estimate the Pst of a new installation based on a statistical analysis of real Pst measurements from similar installations. One criterion for estimating the range of variation in the electric arc's resistance is presented in Horton et al. (2009) and is based on curves that relate the installation's power factor in terms of the arc's resistance, also considering typical values taken by the real power factor in this type of installation. The functionality of these parameter calibration methods is based on models that work with data and need a massive quantity of real measurements from the plant for correct calibration.

Marulanda et al. (2012) consider some of the arc furnace model parameters presented in Alzate et al. (September, 2010) using one of the most widely used classical parameter estimation techniques in practice: maximum likelihood estimation (MLE). However, this methodology requires transforming the nonlinear differential equation that models the electric arc Acha et al. (1990) into an equivalent linear equation for the parameters of the model being estimated and only allows for the estimation of a subset of parameters for that model.

In this paper, we propose and evaluate a methodology based on neural networks for calibrating the parameters of the three-phase arc furnace model proposed in Alzate et al. (September, 2010). This model was implemented in the Matlab®-Simulink® environment and used to generate a large quantity of voltage and current waveforms for electric arcs using different values for the model's parameters in each simulation. The values of the parameters in each simulation were obtained using Latin hypercube sampling (Wyss & Jorgensen, 1998). The simulation data were used to train a multi-layer neural network whose function is to serve as a deterministic emulator of the dynamic model. Once the neural network was trained, the parameters were calibrated by solving an inverse problem in which the (real) voltages and currents are known, but the parameters of the model that generated said signals are unknown. The results obtained were validated by comparing the effective values of the real signals and the simulated signals with the parameters obtained by solving the inverse problem.

This article is organized as follows: section 2 presents a description of the arc furnace model, the neural network used, the backpropagation algorithm, and the method for data sampling; section 3 describes the methodology used to generate the training data for the neural network and carry out its inversion; finally, the results obtained are shown and the study's conclusions are presented.

2. THEORETICAL FRAMEWORK.

This section presents the three-phase arc furnace model and describes the neural networks and the backpropagation algorithm. It also describes the inverse problem and the Latin hypercube method for data sampling.

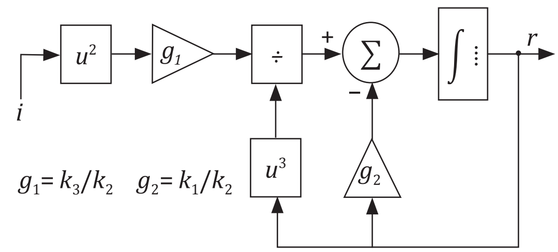
2.1. Mathematical model of a three-phase electric arc furnace

The arc furnace model used to estimate its parameters is presented in Marulanda et al. (2012), so in this study we will give only a short description of it. The model is divided into two parts. Initially, the nonlinear voltage-current characteristic typical of electric arcs is modeled, and then the variable nature of the arc's longitude is considered, moduLating the range of the arc's radius with three low-frequency signals: a sinusoidal signal, a chaotic signal, and a random signal with a Gaussian probability distribution. This is done to simulate the fluctuations that are observed in real voltage and current waveforms in an electric arc furnace. The nonlinear voltage-current characteristic of an electric arc is obtained by solving the following differential equation (Acha et al., 1990):

$$k_1 r^2 + k_2 r \frac{dr}{dt} - k_3 \frac{i^2}{r^2} = 0, \qquad (1)$$

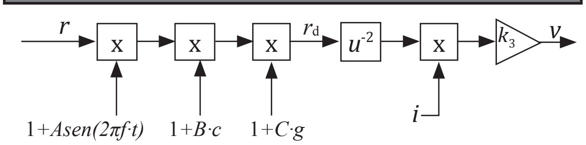
in which r is the radius of the electric arc, i is the arc's instantaneous current, and k_1, k_2 and k_3 are parameters related to the electrical power converted into heat by the arc. **Figure 1** shows a block diagram of how to obtain r by taking current i as input.

Figure 1. Solution of the nonlinear differential equation in block diagram (1).



In the three-phase arc furnace model, we must obtain a respective value of r for each line current. Once r has been determined (by phase), the second part of the model determines the electric arc's dynamic voltage. To do so, we first modulate the range of r with the tree signals, that is, the sinusoidal signal, the chaotic signal, and the random signal with Gaussian distribution. **Figure 2** shows the implementation of the model's second phase in a block diagram.

Figure 2. Block diagram for obtaining the electric arc's voltage.

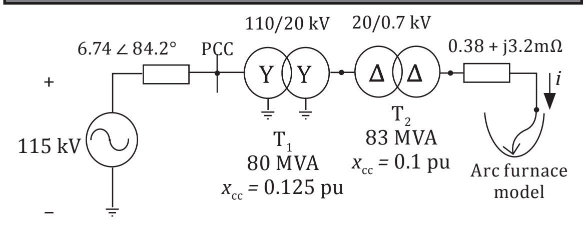


In the figure above, c is a low-frequency chaotic signal generated by Chua's oscillator (Kennedy, 1993) and g is a random signal with Gaussian probability distribution (Manchur, 1992). The constants A, B, and C represent the range modulation indices for the three moduLating signals. Once r_d, the electric arc's dynamic voltage by phase, is obtained, it is determined by the following equation:

$$v = \frac{k_3}{r^2}\, i. \qquad (2)$$

We have used a typical topology for the electrical circuit that feeds the arc furnace (Montanari et al., 1994). **Figure 3** shows the one-line diagram of the circuit, indicating the values used for its components.

Figure 3. One-line diagram of the arc furnace's electrical circuit



In summary, the parameters to be tuned for the arc furnace model using a neural network are k_1, k_2, k_3, A, B, C and f for each phase, a total of 21 parameters..

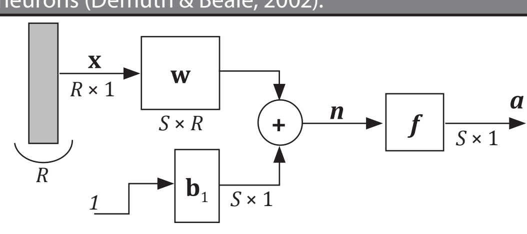
2.2. Neural network

Artificial neural networks are parallel systems for learning and automatically processing information, emuLating the way in which biological neural networks function in the human brain.

2.2.1. Representation of the neural network

The neural networks used in the majority of applications are organized in layers and are totally interconnected (Hilera González & Tome Garcia, 1995), (Rumelhart et al., 1986). These conditions make it possible to create a type of simplified graphic notation which does not explicitly show the neurons, but rather the layers of the network as elements of constructive blocks (Quintero, 2004). For the case of a layer of neurons, the simplified graphic notation is that shown in **Figure 4** (Demuth & Beale, 2002).

Figure 4. Simplified representation of a layer of neurons (Demuth & Beale, 2002).



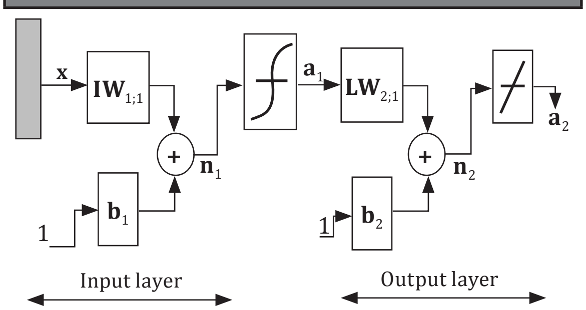
Due to the total connection of input signals x_i with the respective neurons, the number of synaptic weights for each neuron is equal (in dimension), and it is therefore possible to group said weights in a **W** matrix, called the synaptic weights matrix.

It is important to note that if an input is not connected to a certain neutron, the matrix notation is still consistent with the condition that the corresponding weight of said connection is equal to zero. Further, in the majority of applications the activation function of the input layers and the hidden layers is the same for all the neurons in the respective layer. They can therefore be unified in a single function block labeled ***f***. The synaptic weights of a given neuron with the respective inputs xi from the layer can be seen as a row vector, and therefore, all the weights in a layer can be represented by a weights matrix **W**.

$$\mathrm{w} = \begin{bmatrix} w_{11} & \mathrm{K} & w_{1R} \\ \mathrm{M} & \mathrm{O} & \mathrm{M} \\ w_{S1} & \mathrm{L} & w_{SR} \end{bmatrix}, \tag{3}$$

in which S is the number of neurons in the respective layer and R is the number of (scalar) inputs to the layer. In addition to creating a more structural and generalized notation, matrix notation is applied in order to make a distinction between the input layer's synaptic weights matrices and the connections between intermediate layers and the other layers (hidden, output). It is also used to indicate the beginning and end point of the connection between layers. By way of example, the notation $\mathbf{IW}_{1;1}$ is the synaptic weights matrix of the input layer of the neural network, and $\mathbf{LW}_{2;1}$ is the synaptic weights matrix that connects the second layer of the network with the first. This notation is illustrated in **Figure 5** (Demuth & Beale, 2002).

Figure 5. Neural network of two layers with abbreviation notation (Demuth & Beale, 2002)



In accordance with the figure above, the mathematical model of the output function a_2 of the neural network is described by the following equation:

$$a_2 = f\,(\mathrm{W}, \mathrm{x}) = p\,(\mathrm{LW}_{2;1} \cdot t\,(\mathrm{IW}_{1;1} \cdot \mathrm{x} + \mathrm{b}_1) + \mathrm{b}_2), \tag{4}$$

in which $p(z)$ is the linear activation function defined as $purelin(z) = z$ and $t(z)$ is the hyperbolic tangent activation function defined as (Demuth & Beale, 2002)

$$t\,(z) = \frac{2}{1+e^{-2z}} - 1. \tag{5}$$

2.3. The backpropagation algorithm.

The backpropagation algorithm is a classical rule for training neural networks with more than one hidden layer (Rumelhart et al., 1986). The basic idea of training a neural network consists of finding the parameters **IW**, **LW**, $\mathbf{b_1}$ and $\mathbf{b_2}$, that best fit a set of input and output data.

Backpropagation network training consists of a propagation cycle in two phases. Initially, an input example is applied as a stimulus for the network's input layer of neurons and is propagated to the remaining layers in the network's architecture (hidden networks), generating a response in the network's output layer. The responses obtained in the output layer's neurons are then compared with the desired output, that is, with the output pattern that corresponds to the input stimulus. To finish the first phase, an error is calculated for each of the neurons in the output layer.

The algorithm's second phase consists of propagating the error calculated in the initial phase from the output layer toward all the neurons in the hidden layers that directly contribute to the output. These intermediate layers are assigned an error rate based on the contribution of these intermediate neurons to the output obtained in phase 1. This process is repeated in all the network's layers until all the neurons in the network have been assigned an error that describes their relative contribution to the total error in the output. Based on the error received, the synaptic weights of each neuron in the network are modified. It is expected that for a known input stimulus, the network's response will coincide with the desired output (Hilera González & Tome Garcia, 1995).

2.4. The problem of inversion of feedforward networks

A trained neural network can be considered a nonlinear map from the input space to the output space (Bao et al., 1999). Once the neural network has been trained on the set of training data, all the synaptic weights (including biases) in the network remain fixed. Thereby, the assignment of the input space with the output space is known. This assignment is known as feedforward mapping. In general, the correlation of feedforward mapping is a relationship of several to one, because each of the desired outputs can correspond to various different training inputs. Feedforward mapping is expressed as follows:

$$y = f\,(W\,;\,x) \tag{6}$$

in which $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$ and $\mathbf{x} = [x_1, x_2, \ldots, x_R]^T$ represent the outputs and respective inputs in the network, **W** denotes the matrix of fixed synaptic weights in the training process, and the function *f* denotes the feedforward mapping defined by the network's architecture. On the other hand, the problem of inverting a previously trained feedforward neural network (also known as network backpropagation) consists of determining the input **x** that produces a certain output response $\mathbf{d} = [d_1, d_2, \ldots, d_m]^T$. These calculated values of **x** are called network inversions or simply inversions. The mapping of the output space to the input space is known as inverse mapping. In recent years, different algorithms for inverting feedforward networks have been proposed. For more information, see (Linden, 1997).

2.5. Formulating the inversion problem as an optimization problem

Once the network has been trained, the problem that arises is finding the inversion of the network that produces output **d**. In order to determine different inversions for a given output, the inverse problem is formulated as an optimization problem

$$\begin{aligned} &\text{minimize}_x = g\,(x) \\ &\text{subject to: } x_{min} \leq x \leq x_{max}, \end{aligned} \tag{7}$$

in which $g(\mathbf{x})$ is the objective function to be minimized, while $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ are vectors whose components are constant values that represent the range of components for input vector **x** to be determined. The objective function of the model proposed in (Jordan & Rumelhart, 1992) is described in the following equation:

$$g(x) = \left\| d - f\,(W\,;\,x) \right\|^2, \tag{8}$$

in which **d** is the output vector or validation and *f* is the mathematical model that describes the previously trained feedforward network. The feedforward network inversion algorithm can be summed up in two generalized steps.

Algorithm 1. Inversion of feedforward neural networks

Requires: *n* sets of training data (**x-y**) and the validation vector **d**.

1: Creating a feedforward model (creating and training a feedforward network) (Ec.(6)).

2: Inverting the feedforward network and resolving the optimization problem (Ec.(7)).

2.6. The Latin hypercube sampling method.

The Latin hypercube sampling method consists of selecting *n* values for each of the *k* components of vector $\mathbf{x} = [x_1, x_2, \ldots, x_R]^T$ in the following way. The range of possible values taken by each component of vector **x** is divided into *m* non-overlapping intervals with a base of equal probability. A value is randomly selected for each of the *m* intervals with regards to probability density. The *m* samples thereby obtained for the component x_1 are combined randomly with the *m* samples from component x_2. These m pairs are again combined with the m values for component x_3 to form *m* triplets. The process continues in this way until *m k*-sets have been formed. It is helpful to think about these samples of each of the *k* components of vector **x** as the formation of a matrix in which each column contains specific values (samples) for each of the components of **x**, which can be used in a computational model.

3. MATERIALS AND METHODS

The section details the methodology used for estimating the parameters of the electric arc furnace model defined in section 2.1. It presents how we obtain the training data for the neuron network based on the model and also introduces the Matlab® native computational tools for training and inverting the neural network.

3.1. Real database

The real data used to calibrate the arc furnace model's parameters were used by (Cano & Tacca, 2005) and consist of measurements of phase voltages in the secondary of transformer T_2 in **Figure 3** and the currents in the electric arc during five (5) cycles with a sampling frequency of 2048 sps (samples per second) taken in the furnace's fusion phase.

3.2. Data for training the neural network

According to several tests, it is presumed that the values of the parameters for the arc furnace model synthesized by the real data fall within the range

$$0{,}85 \cdot x_i \leq x \leq 1{,}15 \cdot x_i \,, \qquad (9)$$

in which the vector **x** with dimensions [21×1] represents the desired model parameters to be calibrated, $\mathbf{x_i}$ is the vector of initial (known) parameters, and the inequality of the equation above is applied to each of the components of vectors **x** and $\mathbf{x_i}$. The parameters for the three-phase arc furnace model are related to the elements of vector **x** as follows:

$$x = [k_\alpha \; k_b \; k_c \; m_\alpha \; m_b \; m_c \; f\,]^T, \qquad (10)$$

in which $\mathbf{k}_a$ is a row vector whose components are the parameters k_1, k_2 and k_3 for phase a and $\mathbf{m}_a$ is a row vector whose components are the modulation indices A, B and C for the same phase. A similar interpretation applies for the other elements of **x** ($\mathbf{k}_b\,\mathbf{k}_c\,\mathbf{m}_b\,\mathbf{m}_c$). Finally, **f** is the row vector whose components are the frequencies of the three phases f_a, f_b and f_c, as is shown in the block diagram of the model by phase of the arc furnace in **Figure 2**. The components of $\mathbf{x_i}$ are summarized in the following table.

Applying the Latin hypercube sampling method around the inequality seen in **Equation 9**, n vectors **x** for training the neural network are generated and grouped in the columns of matrix **X** with dimensions [21×n], in which n is the number of examples (or simulations) presented to the network and 21 is the number of parameters to be calibrated. In order to determine the training patterns for output **y** of the neural network, it is necessary to simulate the arc furnace model n times using vectors **x** to obtain from each simulation the waveforms of the currents in the electrical arc and the phase voltages in the secondary of transformer T_2, which is shown in **Figure 3**. Then, using the short-time Fourier transform (Jaramillo & Lopez, 2007) with windows of 20 ms and an overlap of 37.5%, the spectrum for each of the six simulated signals (three voltage and three current) is determined. In this way, we form a vector of characteristics **y** for each signal, and each of these vectors is grouped in the columns of matrix **Y** to obtain a matrix with dimensions [54×n] to form the output training data. For each window, we obtain the mean absolute value of the corresponding Fourier transform, and this median is used as an output characteristic, that is, as part of matrix **Y.**

Figure 6 summarizes the application of the methodology used to determine the (**X**, **Y**) input-output training data set for the neural network.

This is how we complete the (**X**, **Y**) input-output training set needed to train the neural network. Now we only need to determine the validation data vector **d** with dimensions [54×1], which consists of the spectrum of real waveforms of the voltages and currents in the secondary of transformer T_2 in the same was as during the training. The characteristics of the validation signals are described in section 3.1..

3.3. Training and inversion of the neural network

The neural network is trained using the Neural Network Toolbox with 500 simulations to construct

Table 1. Initial values for parameters of the arc furnace model $\mathbf{x_i}$

	Phase a	Phase b	Phase c
k	[11283 7,5 9,8]	[12067 6,5 8,0]	[9789 5,1 9,8]
m	[0,0140,021 0,0019]	[0,0210,025 0,0024]	[0,0280,024 0,021]
f	14,6	16,4	15,1

the training set (**X, Y**) in order to avoid possible over-training. It is worth noting that for the neural network training, the three-phase arc furnace is considered a unit due to the interrelation between the currents and voltages of the three-phase furnace.

Once the neural network has been trained, the inverse problem is solved by applying **Equations 7** and **8**. Thereby, for a feedforward network with a hidden layer with a sigmoidal activation function and an output layer whose activation function is linear, the optimization problem set out in **Equation 7** is transformed into the following equation:

$$\text{minimizar}_x = g(x) = \left\| d - p\left(LW_{2;1} \cdot t(IW_{1;1} \cdot x + b_1) + b_2\right) \right\|^2, \quad \text{sujeto a}: x_{min} \leq x \leq x_{max}. \qquad (11)$$

in which p is the linear function and t is the sigmoidal function. To solve **Equation 11**, we used the *fmincon* function in the Optimization toolbox, which finds the minimum of a non-linear function with several variables subject to various restrictions using the trust region reflective optimization method (Coleman & Li, 1996). The number of iterations used in the inversion algorithm is based on a tolerance of 1e-6 (if the algorithm does not converge to the tolerance value, the number of iterations for the method is 1000).

4. RESULTS

This section presents the results obtained from the three-phase electric arc furnace model parameter calibration methodology using neural networks and real data. It also includes a comparison of some characteristics of real wave forms (of voltages and currents) and those synthesized by the model.

The tests consisted of training and inverting a neural network with a hidden layer. For each experiment, the number of neurons in this layer was modified. In the first test, 1 neuron was used in the hidden layer. After the neural network training phase, the inverse problem stated in **Equation 11** is solved in order to obtain the parameters for the arc furnace model. Once the arc furnace model parameters have been obtained, the simulation is run. The real waveforms (of voltages and currents) from the plant were then compared with the simulated waveforms. The procedure was then repeated, varying the number of neutrons in the hidden layer, increasing by 5 neurons in each test until a maximum of 50 neurons was reached. The results of the effective errors between the real data and the model-generated data for 40, 45, and 50 neurons are shown in the following table.

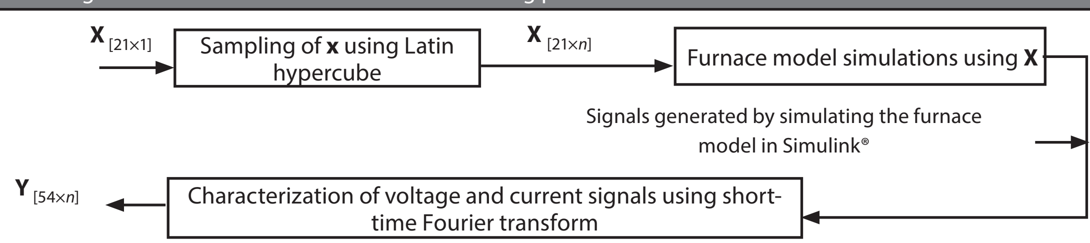
Table 2. Errors obtained for different configurations of the neural network

Neurons	V_a	V_b	V_c	I_a	I_b	I_c
45	1,17	2,79	2,94	0,88	0,08	4,1
50	4,81	10,22	0,41	5,83	13,13	1,74
55	1,65	0,07	0,12	3,35	2,14	2,83

In the table above, we can see that, independent of its topology (the number of hidden layers and the number of neurons per hidden layer), the neural network satisfactorily emulates the non-linear dynamic of the arc furnace. Although the experiments can be done for different configurations in order to reduce the percentage error, the methodology is still valid and independent of the neural network.

In **Table 2** V_a and I_a reference the phase voltage in the secondary of transformer T_2 and the current in the electric arc for phase *a*, respectively. There is no difference for the remaining voltages and currents. The percentage error calculation is obtained based on the following equation:

Figure 6. Diagram that allows us to determine the training pairs for the neural network.

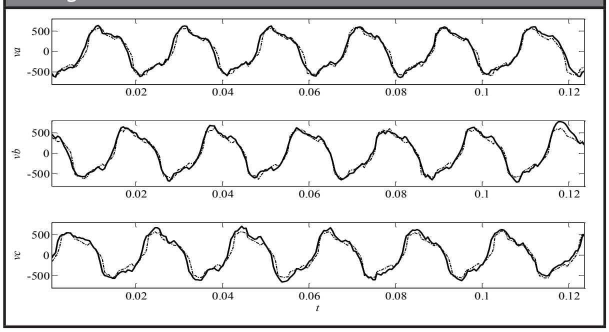


$$e\,(\%) = \left\| \frac{\text{real value} - \text{measured value}}{\text{real value}} \right\| \text{-}100\ \%$$

in which 'real value' references the rms of the real signal, while 'measured value' references the rms of the signals obtained by applying the methodology. The following table shows the values obtained for the components of vector **x** after solving the optimization problem given in **Equation 11** using 45 neurons for the hidden layer.

Based on the results in the table above, the arc furnace model simulation was carried out. The simulation showed that these parameters do not affect the stability of the electrical circuit shown in **Figure 3**. A comparative graph of the real and simulated signals for the phase voltages can be seen in the figure below with a time interval of 0.12 seconds. The real signals correspond to the waves graphed with a solid line, and the simulated signals have been graphed with dotted lines.

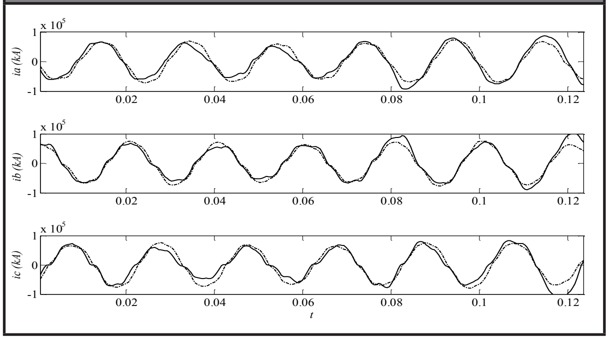
Figure 7. Real and simulate waveforms for the phase voltages of the electric arc furnace



In this figure, we can see that the arc furnace model captures the non-linear nature of the real voltages and also that the values of the parameter vector components obtained with the algorithm allow us to obtain voltage levels similar to real levels. The percentage error obtained for the phase voltages was 1.17% for phase *a*, 2.79% for phase *b*, and 2.94% for phase *c*.

The following figure shows the real and simulated instantaneous currents of the electric arc in each of the phases with a time of 0.12 seconds.

Figure 8. Real and simulated waveforms for the line currents of the electric arc furnace.



In the figure above, we can see that the simulated currents follow the real currents with a high degree of precision in some cycles and that in the remaining cycles, the currents move apart at the positive and negative extremes. The percentage errors obtained for the electric arc currents were 0.88% for phase *a*, 0.08% for phase *b*, and 4.1% for phase c.

5. CONCLUSIONS

Based on the results obtained, we can conclude that the inversion of a neural network applied to the synchronization of arc furnace model parameters gives results similar to the real data from the plant. Care must be taken with the respective configuration of the network (number of layers and number of neurons per layer) due to the strong dependence of calculated errors for voltages and currents in the electric arc on this configuration.

According to the results obtained, the methodology implemented in this study allows us to represent the waveforms of the voltage signals by phase for a real electric arc furnace with a high degree of accuracy. In addition, we obtained a maximum percentage error

Table 3. Results obtained for x by the neural network with real signals.

	Phase a	Phase b	Phase c
k	[10440 7,05 10,28]	[12752 6,63 9,19]	[9525 5,86 9,73]
m	[0,012 0,02 0,0021]	[0,019 0,025 0,0026]	[0,024 0,02 0,002]
f	15,88	16,04	12,83

of 4.1% in the rms for the electric arc currents with regards to the real signals. Finally, it is important to mention that the model approximated the waveforms of the voltage signals better than those of the current due to the lower fluctuation of the voltage signals compared to the strong variations shown in the current waves.

REFERENCES

Acha, E.; Semlyen, A.; Rajakovic, N. (1990). A harmonic domain computational package for nonlinear problems and its application to electric arcs. *IEEE Transactions on Power Delivery*, 5(3), pp.1390-1397.

Alves, M.F.; Assis, Z.M.; Garcia, C.P.; Gomes, D.G. (2010). An Integrated model for the study of flicker compensation in electrical networks. *Electric Power System Research*, 80(10), pp.1299-1305.

Alzate, A.; Marulanda, J.; Escobar, M. (2010). Electric arc furnace modeling for power quality analysis, Bogota, Presented at *IEEE ANDESCON*, 15-17 September, pp.1-6.

Cano, E.A.; Tacca, H.E. (2005). Arc furnace modeling in ATP-EMTP. Presented at the *International Conference on Power System Transients*, Montreal Canada. pp.19-23.

Coleman, T.F.; Li, Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2), pp.418-445.

Collantes, R.; Gómez, T. (1997). Identification and modelling of a three phase arc furnace for voltage disturbance simulation. *IEEE Transactions on Power Delivery*, 12(4), pp.1812-1817.

Demuth, H.; Beale, M.H; Hagan, M.T. (2002). Neural Network Toolbox For Use with MATLAB. User´s Guide. The MathWorks Inc.: 3 Apple Hill Drive. *Natick*, MA 01760-2098.

García-Cerrada, A.; Garcia-González, P.; Collantes, R.; Gómez, T.; Anzola, J. (2000). Comparison of Thyristor Controlled Reactors and Voltaage-Source Inverters for Compensation of Flicker Caused by Arc Furnaces. *IEEE Transactions on Power Delivery*, 15(4), pp.1225-1231.

Hilera, J.R.; García, A. (1995). Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones. 108478971556th ed. RA-MA S.A. Editorial y Publicaciones.

Horton, R.; Haskew, T.A.; Burch, R.F. (2009). A time-domain ac electric arc furnace model for flicker planning studies. *IEEE Transactions On Power Delivery*, 24(3), pp.1450-1457.

Jaramillo, A.; López, R. (2007). Transformada corta de Fourier. *Scientia et Technica*, 8(34), pp.519-521.

Jordan, M.I.; Rumelhart, D.E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3), pp.307-354.

Kennedy, M.P. (1993). Three steps to Chaos-Part I: Evolution. IEEE *Transactions on circuit and systems-I:Fundamental theory and applications*, 40(10), pp.640-656.

Linden, A. (1996). Iterative inversion of neural networks and its applications, in Handbook of Neural Computation, M. *Fiesler and R. Beale. Inst. Phys. Publishing and Oxford* Univ. Press.

Manchur, G. (1992). Development of a model for predicting flicker from electric arc furnace. *IEEE Transactions on Power Delivery,* 7(1), pp.416-426.

Marulanda, J.J.; Sepúlveda, C.D.; Álvarez, M.A. (2012). Estimación de los Parámetros de un Modelo de un Horno de Arco Eléctrico Usando Máxima Verosimilitud. *TecnoLógicas*, (29), pp.69-89.

Montanari, G.C.; Loggini, M.; Cavallini, A.; Pitti, L.; Zaninelli, D. (1994). Arc furnace model for the study of flicker compensation in electrical networks. *IEEE Transactions on Power Delivery*, 9(4), pp.2026-2036.

Ozgun, O.; Abur, A. (2002). Flicker study using a novel arc furnace model. *IEEE Transactions on power delivery*, 17(4), pp.1158-1163.

Quintero, J.J. (2004). Herramienta para Redes Neuronales en Tiempo Real, Tesis profesional como requisito parcial para obtener el título de Ingeniero en Electrónica, Armenia, Universidad del Quindio. Available on: http://www.freewebs.com/jojaqui/tesis2.pdf.

Rumelhart, D.E.; McClelland, J.L.; Group, P.R. (1986). Parallel Distributed Processing. Explorations in the Microstructure. Vol. I ed. Cambridge, Massachusetts, London, England: A Bradford Book.

Wyss, G.D.; Jorgensen, K.H. (1998). A User´s Guide to LHS: Sandia´s Latin Hypercube Sampling Software. Albuquerque, NM 87185-0747: Risk Assessment and System Modeling Department. Sandia National Laboratories.

TO REFERENCE THIS ARTICLE / PARA CITAR ESTE ARTÍCULO / PARA CITAR ESTE ARTIGO /

Álvare-López, M.A.; Baena, C.A.; Marulanda, J.J. (2014). Parameter Calibration for Electric Arc Furnace Models Using Simulation and Neural Networks. Revista EIA, 11(22) July-December, pp. 35-44. [Online]. Available on: http:/dx.doi.org/10.14508/reia.2014.11.22.39-50