

# Artificial neural network for the evaluation of CO<sub>2</sub> corrosion in a pipeline steel

A. Bassam · D. Ortega-Toledo · J. A. Hernandez ·  
J. G. Gonzalez-Rodriguez · J. Uruchurtu

Received: 4 March 2008 / Accepted: 13 May 2008 / Published online: 8 July 2008  
© Springer-Verlag 2008

**Abstract** This paper presents a predictive model for the determination of different types of corrosion by using electrochemical impedance spectroscopy curves and artificial neural network. This proposed model obtains predictions for three different types of corrosion by using Nyquist impedance curves from four input variables: inhibitor concentration, time of exposure, and the real and imaginary experimental component of these curves. The model takes into account the variations of inhibitor concentration over steel to decrease the corrosion rate. For the network, the Levenberg–Marquardt learning algorithm, the hyperbolic tangent sigmoid transfer function and the linear transfer function were used. The best fitting training data set was obtained with five neurons in the hidden layer, which made possible to predict satisfactory efficiency ( $R > 0.99$ ). On the validation of the data set, simulations and theoretical data tests were in good agreement ( $R > 0.9905$ ). The developed model can be used for the determination of the type of curves related to the nature phenomena and rate of corrosion at the metal surface.

**Keywords** Neural network · Corrosion type prediction · Electrochemical impedance

## Introduction

Corrosion is one of the main problems in the oil and gas production and transportation industry. Oil field corrosion

manifests itself in several form, these are “sweet corrosion” generated by carbon dioxide (CO<sub>2</sub>) and/or hydrogen sulfide (H<sub>2</sub>S) called “sour corrosion”, in water injection systems. The most prevalent form of attack found is produced by the presence of oxygen (O<sub>2</sub>) [1]. Inhibitors are currently used to protect against corrosion in all petrochemical facilities in the world because it is cost-effective and flexible. Nitrogen-based organic inhibitors, such as imidazolines or their salts have been successfully used in these applications even without an understanding of the inhibition mechanism [2].

Corrosion inhibition by organic compounds is related to their adsorption properties. Adsorption depends on the nature and the state of the metal surface (microstructure and chemical composition), on the type of corrosive environment, and on the chemical structure of the inhibitor [3]. According to [4], inhibitors incorporate to the corrosion product layer and form a protective barrier between the base metal and the corrosive media. It suggests that the structure of the inhibitor must be appropriate to interact with the corrosion products and that they can be effective on iron carbonates or sulfides, but not effective on oxides. In a previous work [11], carboxyamido-imidazoline was evaluated as corrosion inhibitor in the CO<sub>2</sub> corrosion of pipeline steel; so, the aim of this work is the development, application, and study of the obtained data in this work with a neural network model. This model can be used to determinate the type of impedance spectroscopy curves from different inhibitor concentrations without experimentation. Data evaluation was obtained by using electrochemical impedance spectroscopy (EIS) in an environment containing NaCl and CO<sub>2</sub> to simulate environments found in the transport or crude oil. The impedance spectroscopy curve was obtained to predict corrosion current density values ( $I_{\text{cor}}$ ) with the neural network model; the results are compared with tested experimental data.

A. Bassam · D. Ortega-Toledo · J. A. Hernandez ·  
J. G. Gonzalez-Rodriguez (✉) · J. Uruchurtu  
Universidad Autónoma del Estado de Morelos,  
Centro de Investigación en Ingeniería y Ciencias Aplicadas,  
Av. Universidad 1001, Col. Chamilpa 62209,  
Cuernavaca, Morelos, México  
e-mail: ggonzalez@uaem.mx

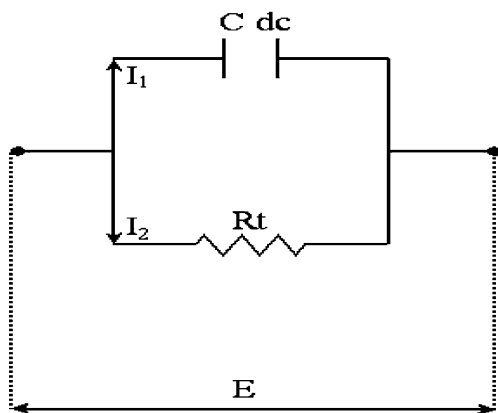
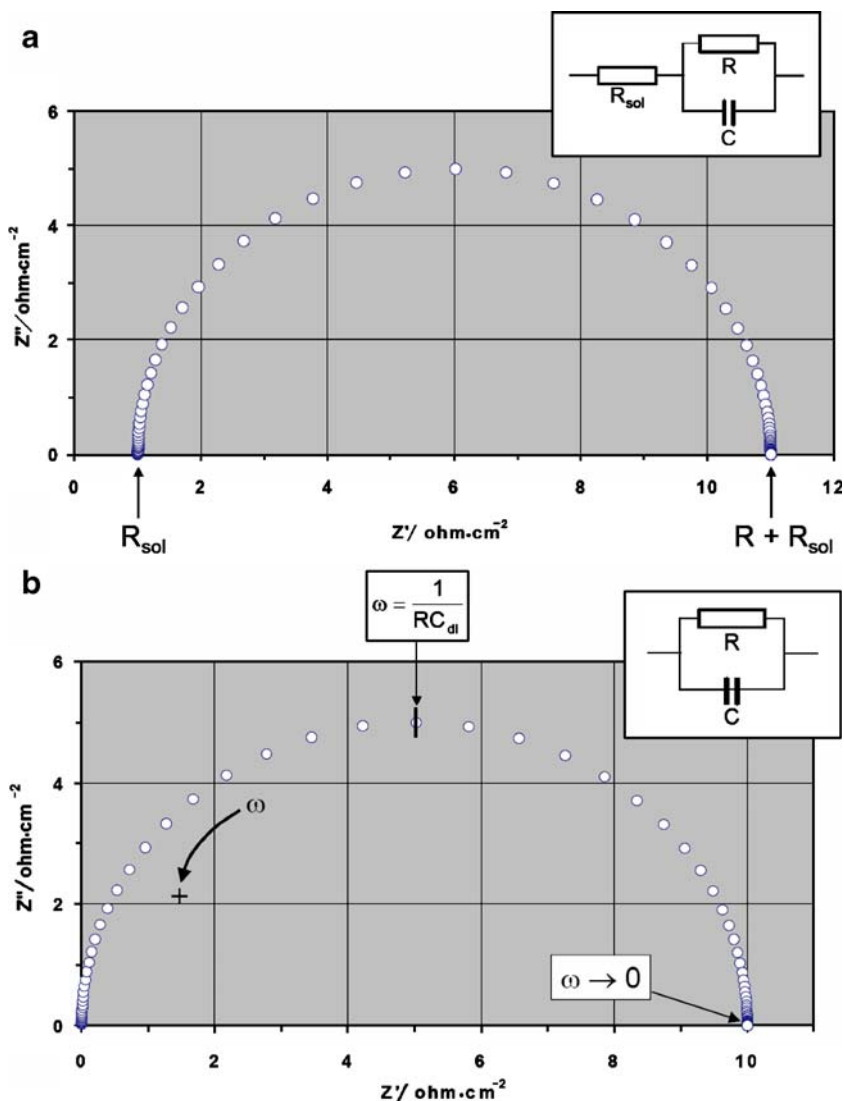


Fig. 1 A simple electric circuit

**Theoretical background**

Electrochemical impedance is a powerful tool used in the measurement and analysis of electrochemical phenomena, including corrosion. A variable AC frequency signal is

Fig. 2 a Nyquist impedance plot and equivalent used electric circuit. b Type 1. charge transfer controlled corrosion impedance plot for metal corrosion



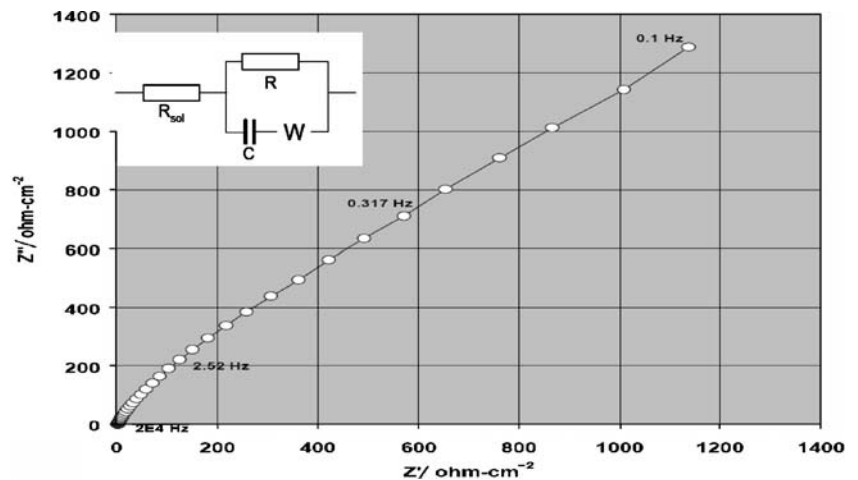
applied to the corrosion system, and a Nyquist or Bode plot is obtained where electrical properties, kinetic and mechanistic as well as electrochemical, and corrosion rate can be derived. Once the experimental data were obtained, attention is drawn in obtaining the parameters by two ways [5], by means of mathematical modeling or through proposed equivalent electric circuits (Fig. 1). Basic parameters can be calculated such as solution resistance, charge transfer resistance, double-layer capacitance, and diffusion-adsorption phenomena among others (Figs. 2b, 3, and 4).

For small applied overpotentials,  $\eta$ , ( $\eta = E - E_{corr}$ ), where  $E_{corr}$  is the open circuit corrosion potential in volts, the net current is proportional to the overpotential:

$$I_{net} = K/R_p \tag{1}$$

where  $R_p$  is the polarization resistance. The corrosion rate of the metal can then be related to  $1/R_p$  by a coefficient, which depends on the mechanism of the corrosion process. This  $R_p$  is one component of the simplest equivalent circuit

**Fig. 3** Type 2. Diffusion controlled corrosion impedance plot for metal corrosion



for the corroding metal. However, for alternating applied voltages, there is another component of the resulting current that must be allowed for, because of the double-layer capacitance  $C_{dc}$  of the surface. Because a current can flow through the interface, causing oxidation or reduction of species, or charge or discharge of the capacitance, these two components are in parallel, and a better equivalent circuit is that shown in Fig. 2a. Also, the effect of solution resistance is added in series to the circuit.

For this circuit, consider how the current varies with potential or, if potential changes with time, how the current varies with time. Consider the voltage as a function of time  $E(t)$  applied to the circuit (Fig. 1). The current flowing through the resistance is given by the previous equation in this case as:

$$I(t) = E(t)/R_p \tag{2}$$

The current is the Faradaic current, related to oxidation or reduction of species, in which the amount of material oxidized or reduced is related to the current by Faradays

laws. The current flowing through the capacitor depends not on the voltage but on the rate of change of the voltage:

$$I(t) = C_{dc}dE(t)/dt \tag{3}$$

And the total current is.

$$I(t) = E(t)/R_t + C_{dc} dE(t)/dt \tag{4}$$

If the rate of change of voltage is small, then the second term is small, and the current is given by the first term such that:

$$R_p = E(t)/I(t) \tag{5}$$

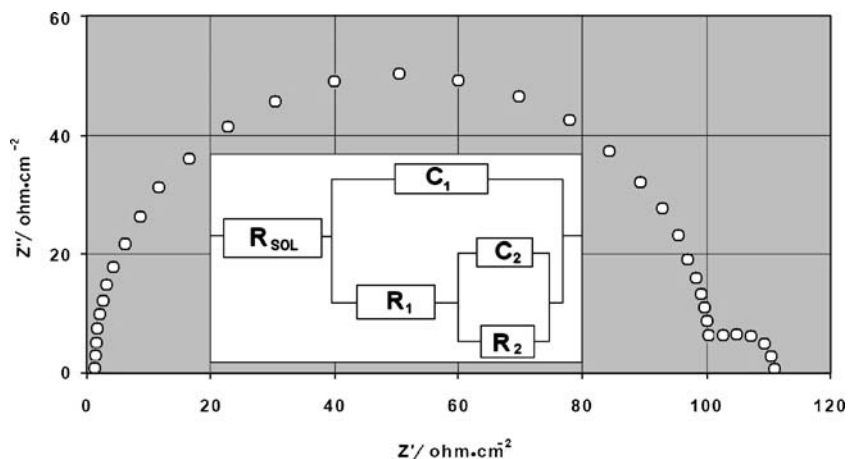
and this is what is obtained with conventional polarization resistance methods with slow potential sweeps, inversely related to the corrosion rate.

When we talk about an AC signal, we mean a sinusoidal signal, and generally the applied signal will be an applied voltage as described by:

$$E(t) = E_0 \sin(\omega t) \tag{6}$$

$$dE(t)/dt = E_0\omega \cos(\omega t) \tag{7}$$

**Fig. 4** Type 3. Impedance plot showing adsorption controlled corrosion effects



**Table 1** Experimental data number used in each type of curve

Type of curve	Type 1	Type 2	Type 3
Experimental data number	585	1,176	1,040

and when

$$I(t) = \frac{E(t)}{R_t} + C_{dc} \frac{dE(t)}{dt} \quad (8)$$

$$I(t) = \frac{E_0}{R_t} \sin(\omega t) + E_0 \omega \cdot C_{dc} \cdot \cos(\omega t) \quad (9)$$

When we plot the current against the time, the first term of the current at (Eq. 9) is in phase with the applied voltage, the second term is 90 ( $\pi/2$  rad) out of phase. The magnitude of both terms is proportional to the magnitude of the applied voltage  $E_0$ .

When we carry out a measurement, it is the total current that is measured, not the two components described above, and it is useful to have a simple expression for the total current we could also express (Eq. 9) as:

$$I(t) = I_0 [\sin(\omega t) \cos \phi + \cos(\omega t) \sin \phi] \quad (10)$$

or

$$I(t) = I_0 \sin(\omega t + \phi) \quad (11)$$

The above equations provide two ways of representing the current response to an applied voltage, as in-phase and out of-phase components or as a modulus and a phase angle. These describe the admittance of the circuit also. We have admittance here because we have measured the current response to voltage. By using the symbol  $Y$  for the admittance, we can write:

$$Y(\omega) = \frac{1}{R_t} + j\omega C_{dc} \quad (12)$$

or

$$Y(\omega) = Y' + j \cdot Y'' \quad (13)$$

where  $Y'$  is referred to as the admittance real component and  $Y''$  as the imaginary component. The impedance  $Z$  is the inverse value of the admittance used to describe the circuit representing the corrosion process at the metal surface. Note that we have described the admittance as  $Y(\omega)$  because it depends on frequency. Generally, but not in this case, both  $Y'$  and  $Y''$  also depend on frequency.  $Y'$  is proportional to the in-phase current component. Here, we need to recognize that  $j$  is equal to the square root of  $-1$ ,

and the expression for the admittance is a complex number and can be manipulated, added, as such.

For the simple circuit in Fig. 4, the admittance is:

$$Y = 1/R + j\omega C = 1 + j\omega C R/R \quad (14)$$

For the impedance, the inverse expression is:

$$Z = RI + 1/j\omega C \quad (15)$$

To convert this to an expression with real and imaginary parts, we remove the  $j$  term from the bottom:

$$Z = [R/1 + \omega^2 C^2 R^2] - [j\omega^2 C^2 R^2/1 + \omega^2 C^2 R^2] = Z' + Z'' \quad (16)$$

To express this term of modulus and phase angle,

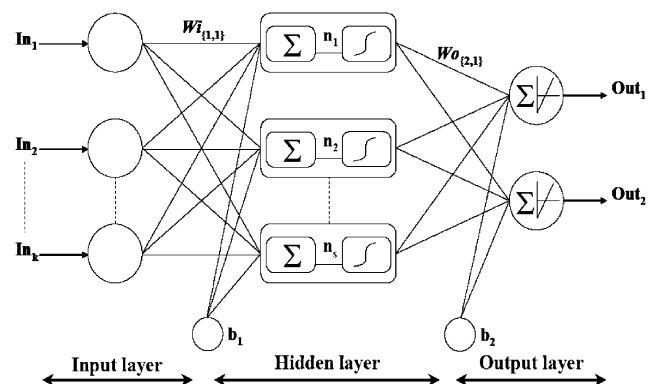
$$|Z| = R / [\omega^2 C^2 R^2]^{1/2} \quad (17)$$

$$\tan \phi = -\omega CR \quad (18)$$

With a conventional impedance measurement, we provide an AC signal given by Eq. (6) and measure as current response given by Eq. 11. Formally, this is an admittance measurement because we look at the current response to an applied voltage. The analyzer processes the data to provide either the impedance modulus  $|Z|$  and phase angle or the real and imaginary components of the impedance related to the corrosion current [6].

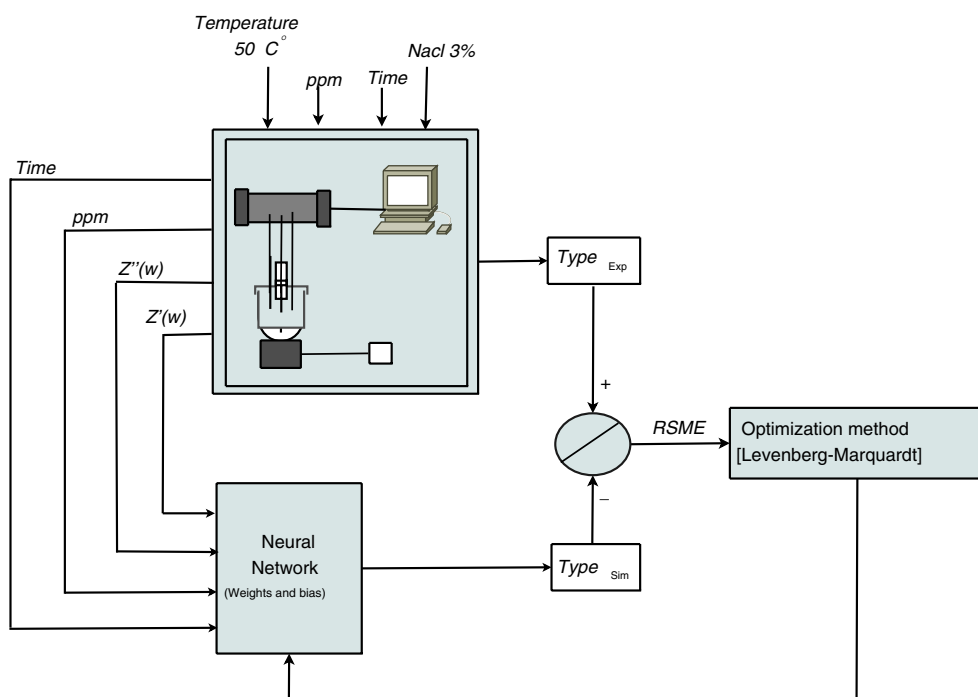
## Experimental data

Experimental data base provided by Gonzalez-Rodriguez et al. [11] consists of different values of inhibitor concentration, time, real component, and imaginary component. Table 1, shows the experimental data number for the three type of curves (experimental total data 2801), which was enough to develop the neural network model.



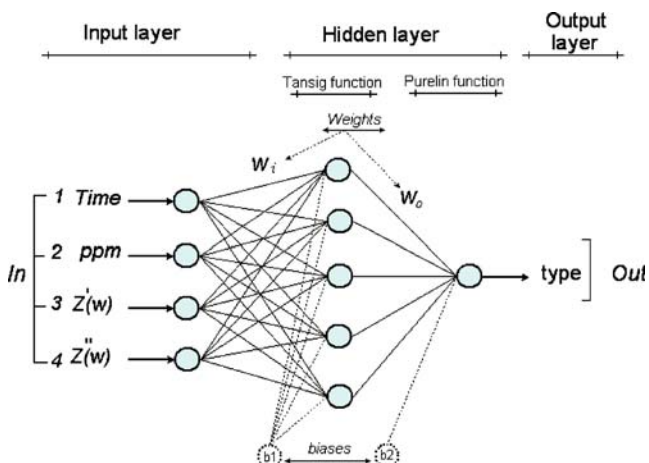
**Fig. 5** The neural network computational model.  $K$  Input number,  $In$  input,  $Out$  output,  $Continuous$  lines weights and bias

**Fig. 6** Recurrent network architecture to obtain the type of corrosion and the procedure used for learning neural network



**Neural network layering**

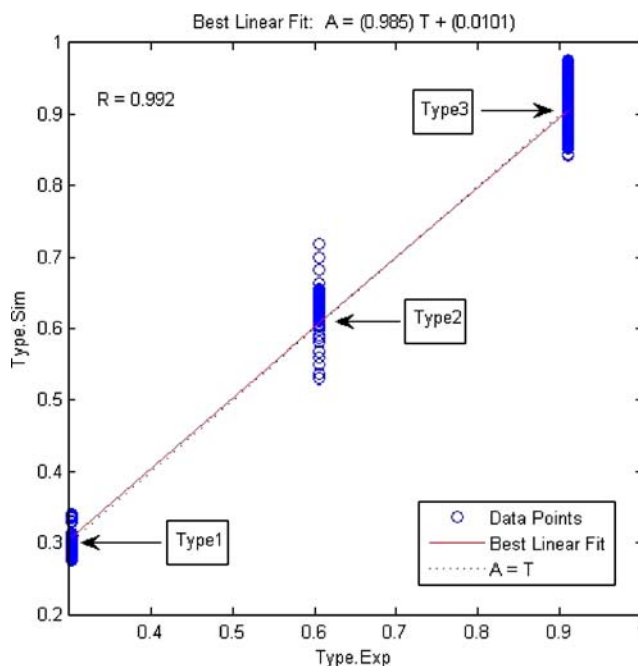
The neurons are grouped into distinct layers and interconnected according to a given architecture. As in nature, the network function is determined largely by the connections between elements (neurons). Each connection between two neurons has a weight coefficient attached to it. The standard network structure for an approximation function is the multiple-layer perception (or feed forward network). The feed forward network often has one or more hidden layers of sigmoidal neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear



**Fig. 7** Proposed neural network model to calculate the type of corrosion

output layer lets the network produce values outside the  $-1$  to  $+1$  range [7]. For the network, the appropriate notation is used in two-layer networks [8]. A simplified sketch of the network’s structure and behavior is presented in Fig. 5.

The number of neurons in the input and output layers is given, respectively, by the number of input and output variables in the process under investigation. In this work,



**Fig. 8** Experimental versus simulated type values by the proposed neural model

the input layer consists of four variables, while the output layer contains one variable for the model (see Fig. 7). The optimal number of neurons in the hidden layer ( $s$ )  $n_s$  is difficult to specify and depends on the type and complexity of the task. This number is usually determined iteratively. Each neuron in the hidden layer has a bias  $b$  (threshold), which is added to the weighted inputs to form the neuron input  $n$  (Eq. 19). This sum,  $n$ , is the argument of the transfer function  $f$ .

$$n = W_{i\{1,1\}}ln_1 + W_{i\{1,2\}}ln_2 + \dots + W_{i\{1,k\}}ln_k + b \quad (19)$$

The coefficients associated with the hidden layer are grouped into matrices  $Wi$  (weights) and  $b1$  (biases). The output layer computes the weighted sum of the signals provided by the hidden layer, and the associated coefficients are grouped into matrices  $Wo$  and  $b2$ . Using the matrix notation, the network output can be given by Eq. 20:

$$\text{Out} = g(Wo \times f(Wi \times \ln + b1) + b2) \quad (20)$$

Hidden layer neurons may use any differentiable transfer function to generate their output. In this work, a hyperbolic tangent sigmoidal transfer function and a linear transfer function were used for  $f$  and  $g$ , respectively [8]. The number of network coefficients (weights and biases) is given by Eq. 21.

$$m = n(\ln + 1) + \text{Out}(n + 1) \quad (21)$$

### Neural network learning

A learning (or training) algorithm is defined as a procedure that consists of adjusting the coefficients (weights and biases) of a network, to minimize an error function (usually a quadratic one) between the network outputs, for a given set of inputs, and the correct (already known) outputs. If smooth nonlinearities are used, the gradient of the error function can be computed by the classical back propagation procedure [9]. Previous learning algorithms used this gradient directly in a steepest descent optimization, but recent results have shown that second-order methods are far more effective. In this work, the Levenberg–Marquardt algorithm optimization procedure—in the Matlab Neural

Network Toolbox [8]—was used. This algorithm is an approximation of Newton's method, which was designed to approach second-order training speed without having to compute the Hessian matrix [10]. Despite the fact that computations involved in each iteration are more complex than in the steepest descent case, the convergence is faster, typically by a factor of 100. The root mean square error (RMSE) is calculated with the experimental values and network predictions. This calculation is used as a criterion for model adequacy (Fig. 6). Experimental database (see Table 1) were split into learning database (50% of experimental data set) and testing database (50% of experimental data set) to obtain a good representation of the situation diversity.

### Results and discussion

The neural network model which was developed (Fig. 7) involved five neurons ( $n_s=5$ ) in the hidden layer (20 weights ( $Wi=20$ ;  $Wo=5$ ) and six biases ( $b1=5$  and  $b2=1$ )) to predict a type of curve of impedance (see Figs. 2b, 3 and 4).

In the learning database given by the root-mean-square error (RSME) trial versus the iteration number, the algorithm was worked out for one to five neurons in the hidden layer. The obtained results (data not shown) proved that the typical learning error decreased when the number of neurons in the hidden layer increased; this is evident as the number of adjusted parameters increased. Nevertheless, one of the problems that occur during feed-forward neural network training is called “over-fitting” [12]. The comparison of the RSME calculated for the learning and testing database is a good criterion to optimize the number of iterations and avoid over-fitting. In this neural network, the RSME showed that for six neurons in the hidden layer, the learning database value was small with respect to the testing database. Then, according to RSME results, the optimal number of neurons in the hidden layer was five. Figure 8 shows the experimental data versus simulated data of the impedance type. We could observe how the simulated data have the expected relationship with respect to experimental data. The residuals were small for all experiments in the learning base ( $R=0.992$ ).

**Table 2** Statistical parameters obtained for the neural network

Parameter	$Wi$				$Wo$	$b1$	$b2$
With $L=1$ , $S=5$ , and $K=4$	0.3003	-32.7024	-7.4404	23.2590	-4.4196	1.7903	0.5509
	57.938	16.2788	7.2504	-5.4220	-7.3929	-5.5939	
	-7.7319	46.0374	5.1389	-13.0085	0.3761	0.4960	
	0.6019	-30.0359	-3.6111	26.4383	4.6466	1.6267	
	-62.5418	-18.8813	-5.8075	6.0983	-7.2973	6.1744	

**Table 3** Comparison between simulated and experimental data

Test A						Test B						Test C					
Input parameters				Type of curve		Input parameters				Type of curve		Input parameters				Type of curve	
Time (h)	ppm	Z'(w)	Z''(w)	Exp	Sim	Time (h)	ppm	Z'(w)	Z''(w)	Exp	Sim	Time (h)	ppm	Z'(w)	Z''(w)	Exp	Sim
4	0	1.19069	0.0322	1	1.3062	4	25	68.8801	-44.3194	2	1.9952	6	10	1.60271	-0.26547	3	3.0254
4	0	1.18535	0.02948	1	1.3062	4	25	75.0442	-48.6856	2	1.9952	6	10	1.62703	-0.30568	3	3.0254
4	0	1.16405	0.00937	1	1.3062	4	25	82.368	-53.3533	2	1.9952	6	10	1.66239	-0.35331	3	3.0253
4	0	1.16717	-0.0051	1	1.3062	4	25	89.7784	-58.094	2	1.9952	6	10	1.69813	-0.40329	3	3.0253
4	0	1.15947	-0.0252	1	1.3062	4	25	97.83	-62.9574	2	1.9952	6	10	1.76257	-0.46524	3	3.0252
4	0	1.16288	-0.0421	1	1.3062	4	25	106.563	-67.5201	2	1.9952	6	10	1.81581	-0.52069	3	3.0252
4	0	1.16742	-0.0595	1	1.3062	4	25	116.286	-73.3311	2	1.9952	6	10	1.87578	-0.58371	3	3.0251
4	0	1.21354	-0.0801	1	1.3062	4	25	127.053	-77.8159	2	1.9952	6	10	1.9499	-0.65828	3	3.0251
4	0	1.22558	-0.1041	1	1.3062	4	25	137.922	-84.4778	2	1.9952	6	10	2.03073	-0.73819	3	3.025
4	0	1.2263	-0.1292	1	1.3062	4	25	150.129	-90.6962	2	1.9952	6	10	2.131	-0.82847	3	3.0249
4	0	1.23698	-0.1561	1	1.3062	4	25	161.155	-96.317	2	1.9952	6	10	2.2389	-0.92447	3	3.0248
4	0	1.25409	-0.1877	1	1.3062	4	25	174.893	-103.18	2	1.9952	6	10	2.34984	-1.04208	3	3.0247
4	0	1.27317	-0.226	1	1.3062	4	25	189.33	-112.256	2	1.9952	6	10	2.47825	-1.15676	3	3.0246
4	0	1.28476	-0.2695	1	1.3062	4	25	205.131	-121.399	2	1.9952	6	10	2.6151	-1.30016	3	3.0245
4	0	1.31001	-0.3226	1	1.3062	4	25	221.87	-130.212	2	1.9952	6	10	2.78729	-1.45342	3	3.0244
4	0	1.33582	-0.3807	1	1.3062	4	25	238.993	-139.908	2	1.9952	6	10	2.96205	-1.62667	3	3.0242
4	0	1.3717	-0.455	1	1.3062	4	25	258.27	-152.242	2	1.9952	6	10	3.17017	-1.81633	3	3.0241
4	0	1.37994	-0.5294	1	1.3062	4	25	279.675	-161.144	2	1.9952	6	10	3.37963	-2.03329	3	3.0239
4	0	1.39774	-0.6202	1	1.3062	4	25	301.582	-175.126	2	1.9952	6	10	3.57959	-2.26034	3	3.0237
4	0	1.45495	-0.735	1	1.3062	4	25	324.669	-190.952	2	1.9952	6	10	3.8486	-2.53422	3	3.0234
4	0	1.52727	-0.8689	1	1.3062	4	25	353.817	-206.816	2	1.9952	6	10	4.18533	-2.84758	3	3.0232
4	0	1.61435	-1.0221	1	1.3062	4	25	381.423	-221.914	2	1.9951	6	10	4.49418	-3.19229	3	3.0229
4	0	1.71127	-1.2124	1	1.3061	4	25	414.538	-239.526	2	1.9951	6	10	4.83873	-3.58914	3	3.0225
4	0	1.84323	-1.4242	1	1.3061	4	25	445.404	-258.26	2	1.9951	6	10	5.17392	-4.06298	3	3.0221
4	0	1.98184	-1.6678	1	1.3061	4	25	481.214	-278.066	2	1.9951	6	10	5.57537	-4.61766	3	3.0216
4	0	2.14409	-1.9467	1	1.3061	4	25	519.618	-299.251	2	1.9951	6	10	6.00105	-5.27948	3	3.021
4	0	2.34103	-2.2711	1	1.3061	4	25	560.624	-322.452	2	1.9951	6	10	6.50924	-6.07932	3	3.0203
4	0	2.6268	-2.6497	1	1.306	4	25	603.857	-347.367	2	1.9951	6	10	7.05344	-7.03615	3	3.0194
4	0	2.88799	-3.081	1	1.306	4	25	650.211	-375.101	2	1.9951	6	10	7.68975	-8.18373	3	3.0184
4	0	3.19819	-3.5829	1	1.306	4	25	699.565	-405.701	2	1.995	6	10	8.4007	-9.5541	3	3.0171
4	0	3.55638	-4.1683	1	1.3059	4	25	752.547	-439.757	2	1.995	6	10	9.25896	-11.2	3	3.0156
4	0	3.95513	-4.8479	1	1.3059	4	25	809.301	-478.089	2	1.995	6	10	10.2856	-13.1708	3	3.0138
4	0	4.43189	-5.6386	1	1.3058	4	25	870.663	-521.088	2	1.995	6	10	11.5641	-15.476	3	3.0118
4	0	4.99349	-6.5642	1	1.3058	4	25	937.133	-569.37	2	1.995	6	10	13.1455	-18.2373	3	3.0093
4	0	5.61	-7.6066	1	1.3057	4	25	1,007.78	-622.16	2	1.995	6	10	15.0476	-21.3383	3	3.0065
4	0	6.42004	-8.8906	1	1.3056	4	25	1,090.73	-686.069	2	1.9949	6	10	17.6576	-25.1304	3	3.0031
4	0	7.32152	-10.368	1	1.3055	4	25	1,178.64	-754.169	2	1.9949	6	10	20.8836	-29.335	3	2.9995
4	0	8.32633	-11.99	1	1.3054	4	25	1,273.09	-828.783	2	1.9949	6	10	24.7732	-33.9017	3	2.9955
4	0	9.5749	-14.002	1	1.3052	4	25	1,383.94	-917.23	2	1.9949	6	10	30.0148	-39.0752	3	2.991
4	0	11.0646	-16.312	1	1.3051	4	25	1,506.86	-1013.43	2	1.9948	6	10	36.5113	-44.2759	3	2.9867
4	0	12.7254	-18.854	1	1.3049	4	25	1,637.66	-1117.66	2	1.9948	6	10	44.1092	-48.8816	3	2.983
4	0	14.8282	-21.938	1	1.3047	4	25	1,794.44	-1232.04	2	1.9948	6	10	53.42	-53.7303	3	2.9791
4	0	17.1581	-25.318	1	1.3045	4	25	1,958.98	-1353.37	2	1.9948	6	10	63.5802	-57.2269	3	2.9766
4	0	20.0881	-29.36	1	1.3042	4	25	2,144.97	-1482.36	2	1.9947	6	10	74.9022	-59.2725	3	2.9755
4	0	23.4369	-33.835	1	1.3039	4	25	2,364.58	-1629.69	2	1.9947	6	10	86.5954	-59.2791	3	2.9762
4	0	27.5843	-38.814	1	1.3035	4	25	2,589.62	-1778.12	2	1.9947	6	10	97.689	-57.7069	3	2.9784
4	0	32.4631	-44.426	1	1.3032	4	25	2,845.81	-1926.77	2	1.9947	6	10	108.345	-55.1325	3	2.9815
4	0	38.2065	-50.706	1	1.3028	4	25	3,114.89	-2089.37	2	1.9946	6	10	117.299	-51.6464	3	2.9852
4	0	45.0989	-57.486	1	1.3023	4	25	3,411.08	-2266.31	2	1.9946	6	10	126.08	-48.2906	3	2.9889
4	0	53.3012	-64.969	1	1.3019	4	25	3,721.6	-2441.62	2	1.9946	6	10	131.924	-43.6166	3	2.9936
4	0	62.8789	-72.987	1	1.3014	4	25	4,066.65	-2639.27	2	1.9946	6	10	138.432	-40.9049	3	2.9965
4	0	74.1556	-81.044	1	1.301	4	25	4,414.36	-2854.65	2	1.9946	6	10	143.067	-36.8351	3	3.0006
4	0	87.3862	-89.181	1	1.3006	4	25	4,800.5	-3072.53	2	1.9946	6	10	147.361	-34.5326	3	3.003
4	0	102.7	-96.865	1	1.3004	4	25	5,206.11	-3332.49	2	1.9946	6	10	152.705	-33.0706	3	3.0047
4	0	120.072	-103.51	1	1.3003	4	25	5,656.77	-3620.83	2	1.9946	6	10	157.531	-30.6655	3	3.0073
4	0	139.442	-108.27	1	1.3006	4	25	6,153.85	-3902.13	2	1.9946	6	10	162.774	-28.5193	3	3.0097
4	0	160.418	-110.58	1	1.3011	4	25	6,642.91	-4242.41	2	1.9945	6	10	167.346	-24.9825	3	3.0133
4	0	182.459	-110.08	1	1.3021	4	25	7,244.25	-4651.69	2	1.9945	6	10	172.055	-21.863	3	3.0165
4	0	204.413	-106.34	1	1.3034	4	25	7,839.14	-5064.88	2	1.9945	6	10	175.237	-18.5839	3	3.0198
4	0	225.437	-98.519	1	1.3051	4	25	8,547.84	-5508.44	2	1.9945	6	10	176.893	-14.0106	3	3.0242
4	0	244.603	-88.314	1	1.3069	4	25	9,285.3	-6004.4	2	1.9945	6	10	177.41	-11.1774	3	3.0269
4	0	261.12	-75.859	1	1.3089	4	25	1,0111.4	-6552.77	2	1.9945	6	10	180.188	-7.67294	3	3.0304

According to the obtained model, Table 2 gives the optimum coefficients ( $W_i$ ,  $W_o$ ,  $B_1$  and  $B_2$ ) of the best fit of the model for five neurons in the hidden layer. These coefficients are used to validate the model with fresh data (testing data). It must be clarified that this model is done to predict types of corrosion but not corrosion rates. To carry out this, it is necessary to train the neuronal network to measure or predict the charge transfer resistance,  $R_{ct}$ , and use Eq. 1, where  $R_p$  is now  $R_{ct}$ .

### Model validation

The model thus calculated was then validated using the values of test database. Table 3 shows three tests with different input values and compares experimental and simulated types of curves. The neural model fitted these unknown data well, including the three types of curves, which are complex and very difficult to predict. In the first test, the simulated values were about 1.3, which was considered the curve of corrosion type 1. To the other test (type 2–3) the simulated values were expected values with respect to experimental corrosion type. This difference for simulated values for type 1 and 2–3 is caused by the experimental data number considered for each type of curves (see Table 1). The residuals in the testing database were small and their distribution well-balanced (data not shown). The regression coefficient,  $R=0.99$ , confirmed this satisfactory agreement. Furthermore, the test of lack of fit (LOF) [13] was used to confirm the dispersion inherent in the data. As a result of this test, the  $F$ -statistic value was 172.78, and according to hypotheses for an  $\alpha=0.01$ , the  $F^*>6.366$ . We reject the null hypothesis in favor of the alternative which is lack of linear fit [13]. Consequently, there is sufficient evidence at the  $\alpha=0.01$  level to conclude that there is lack of linear fit. The model thus developed was able to predict the type of corrosion as a function of process parameters throughout the experimental domain.

### Conclusions

This study proposed a neural network model that can predict the corrosion type in a pipeline steel as a function of four input parameters and exhibit a good ability for

generalization. The four input parameters used to determine simulation of corrosion type were the inhibitor concentration, experimental time, real experimental component, and imaginary experimental component. The neural network model was successfully trained with experimental database and validated with a fresh database (in the specified range of key operating conditions).

The model thus developed may enable the implementation of smart sensors for online quality determination (corrosion type) in pipeline steel. Corrosion type can be computed from inhibitor concentration, time, real, and imaginary component. The validity of the corrosion type thus computed will be confirmed by comparing the measured and computed output values. The computation of corrosion type using the neural model is then possible to halt the process when the target corrosion type has been attained.

**Acknowledgements** We thank the SRE (Secretaria de Relaciones exteriores de México) for the economic support received for the development of this work. We also thank Dr. Jorge Alberto Andaverde Arredondo for the statistical support for this work.

### References

1. Kermani MB, Morshed A (2003) Corrosion 59:659
2. Xueyuan Z, Fengping W, Yufang H, Yuanlong D (2001) Corros Sci 43:1417
3. Bentiss F, Lgrenee M, Traisnel M, Hornez JC (1999) Corros Sci 41:789
4. Rosenfled IL, Bongomolov DB, Gorodetskii AE, Kazanskii LP, Frolova LV (1982) Zashita Metalov 18:163
5. Cottis R, Turgoose S (1999) In: Syrett BC (ed) Electrochemical impedance and noise, corrosion testing, easy. Nace internacional, USA
6. Gonzalez\_Fernandez JA (1989) Control de la Corrosión Estudio y Medida por Técnicas Electroquímicas. C.SIC, Spain
7. Limin F (ed) (1994) Neural networks in computer intelligence. McGraw-Hill International Series in Computer Science
8. Demuth H, Beale M (eds) (1998) Neural network toolbox for Matlab-user's guide version 3. Natick, MA, The MathWorks Inc
9. Rumelhart DE, Hinton GE, Williams RJ (1986) Parallel data Processing (1):318
10. Martin T, Hagan MT, Mohammad BN (1994) IEEE Trans Neural Netw 6:989
11. Gonzalez-Rodriguez JG, Zeferino-Rodriguez T, Ortega DM, Serna S, Campillo B, Casales M, Valenzuela E, Juarez-Islas J (2007) Int J Electrochem Sci 2:883
12. Hernández-Pérez JA, García-Alvarado MA, Trystram G, Heyd B (2004) Innovative Food Science and Emerging Technologies 5:57
13. Neter J, Kunter MH, Nachtsheim CJ, Wsserman W (1996) Applied linear statistical models, 4th edn. WCB, McGraw-Hill, USA