PERGAMON

Corrosion Science 43 (2001) 1373–1392

CORROSION SCIENCE

www.elsevier.com/locate/corsci

Probabilistic modelling of CO_2 corrosion laboratory data using neural networks

Srdjan Nesic [a,*], Magnus Nordsveen [b], Nigel Maxwell [a], Miran Vrhovac [c]

[a] *Department of Mechanical Engineering, The University of Queensland, Brisbane, Qld 4072, Australia*
[b] *Institute for Energy Technology, P.O. Box 40, 2007 Kjeller, Norway*
[c] *Research, Development and Training Department, ISQ – Instituto de Soldadura e Qualidade, Portuguese Welding Institute, 2781 Oeiras-Codex, Portugal*

Received 27 March 2000; accepted 6 October 2000

Abstract

The present paper addresses two major concerns that were identified when developing neural network based prediction models and which can limit their wider applicability in the industry.

The first problem is that it appears neural network models are not readily available to a corrosion engineer. Therefore the first part of this paper describes a neural network model of CO_2 corrosion which was created using a standard commercial software package and simple modelling strategies. It was found that such a model was able to capture practically all of the trends noticed in the experimental data with acceptable accuracy. This exercise has proven that a corrosion engineer could readily develop a neural network model such as the one described below for any problem at hand, given that sufficient experimental data exist. This applies even in the cases when the understanding of the underlying processes is poor.

The second problem arises from cases when all the required inputs for a model are not known or can be estimated with a limited degree of accuracy. It seems advantageous to have models that can take as input a range rather than a single value. One such model, based on the so-called Monte Carlo approach, is presented. A number of comparisons are shown which have illustrated how a corrosion engineer might use this approach to rapidly test the sensitivity of a model to the uncertainities associated with the input parameters.

Keywords: CO_2 corrosion; Modelling; Neural networks; Monte Carlo method; Probabilistic approach

* Corresponding author. Tel.: +61-7-3365-3668; fax: +61-7-3365-4766.
E-mail address: secretary@mech.uq.edu.au (S. Nesic).

0010-938X/01/$ - see front matter
PII: S0010-938X(00)00157-8

1. Introduction

Prediction of corrosion rates is difficult. The complexity of the underlying physico-chemical phenomena is often such that our understanding is significantly below a level required for mechanistic modelling. The heterogeneous electrochemical processes underlying corrosion are difficult to capture with a mechanistic model, even without the additional complications commonly introduced by environmental, metallurgical and mechanical aspects of the problem. Therefore, researchers have in the past often resorted to empirical modelling of various complexities. In this paper the discussion is going to be narrowed down to CO_2 corrosion of carbon steel, although most of the methodology and argumentation applies to a broader field of corrosion modelling.

A thorough review of the field of CO_2 corrosion prediction has recently been published [1]. The numerous models of CO_2 corrosion were grouped into three categories: *mechanistic*, *semi-empirical* and *empirical* models, based on how firmly they were grounded in theory. Since the mentioned review a number of new modelling studies have appeared [2–5], following much of the same approaches as outlined in the review. The performance of different models, covering all three groups, was recently tested by comparing the predictions with a large experimental CO_2 corrosion database. The fully empirical model based on the neural network approach performed significantly better than the others while the fully mechanistic model was worst in this comparison.

After this benchmarking exercise which has highlighted neural network based modelling, a number of possible pathways were explored for their further development and application to prediction of CO_2 corrosion. As a result, a hybrid model was created recently which combines the reliability of a mechanistic model with the flexibility of the neural network approach. Results of this study will appear soon. The present paper addresses two major concerns which were identified when developing the neural network prediction models and which can limit their more general applicability in the industry.

(1) The existing neural network model of CO_2 corrosion [6] owes its prediction success, in the first place, to the sophisticated design and training strategies which were used in model development. For example, the evolved descriptor approach implemented through Fourier series "closed" many gaps in the empirical database, while the genetic algorithm, which guided the neural network training process, avoided pattern memorization. The details of this neural network model are explained elsewhere [6]. The obvious shortcoming of this approach is that most of the advanced modelling techniques which were employed there are not readily available to a corrosion engineer who might be interested in creating her/his own neural network model for the corrosion process at hand, without having to become a neural network expert. Therefore this paper describes an attempt to create a successful neural network model of CO_2 corrosion using a standard commercial software package and simple modelling strategies. Indeed any other process (be it corrosion or not) where a large enough reliable database exists, could have been used in this exercise.

(2) When attempting to use a particular model, one typically has to provide all the required inputs before an output can be found. This poses a problem to a user who might not know all the required inputs or can estimate them only with a limited degree of accuracy. Therefore it seems advantageous to have a model which can take a range (distribution, real or guessed) of values rather than a single value as input. In the second part of the present paper one such model based on the so-called Monte Carlo approach is presented. The neural network model of CO_2 corrosion models developed in this study was used to illustrate this approach, however any other model could have been used within the same Monte Carlo model framework.

Only in a handful of cases neural networks were used in the past to predict corrosion behaviour. The risk of stress corrosion cracking [7], onset of crevice corrosion [8], interpretation of polarization scans [9], and in the most recent paper, atmospheric corrosion [10], were successfully modelled using this technique. In most of these studies the neural networks outperformed the more traditional curve-fitting techniques.

In the following sections, a brief background is first presented for the three cornerstones of the present work: CO_2 corrosion, neural networks and the Monte Carlo method, followed by the presentation of the models and their testing.

2. Theoretical background

2.1. CO_2 corrosion

Aqueous CO_2 corrosion of carbon steel is an electrochemical process involving the anodic dissolution of iron and the cathodic evolution of hydrogen. The overall reaction is:

$$Fe + CO_2 + H_2O \rightarrow FeCO_3 + H_2 \quad (1)$$

The electrochemical reactions are often accompanied by the formation of solid films of $FeCO_3$ (and/or Fe_3O_4) which can be protective or non-protective depending on the conditions under which they are formed. One of the most important individual reactions is the anodic dissolution of iron:

$$Fe \rightarrow Fe^{2+} + 2e^- \quad (2)$$

It is believed that the presence of CO_2 increases the rate of corrosion of mild steel in aqueous solutions by increasing the rate of the hydrogen evolution reaction. The presence of H_2CO_3 enables hydrogen evolution at a high rate even at pH > 5. Thus at a given pH the presence of CO_2 leads to a higher corrosion rate than would be found in a solution of a strong acid. It is not known whether H_2CO_3 is reduced directly (as assumed by many workers in the field [12,14]) or that it serves as an extra source of H^+ ions. Many have assumed that the two reactions are independent and the net cathodic current is the sum of the currents for the two reactions [14,17]:

$$2H^+ + 2e^- \rightarrow H_2 \quad (3)$$

$$2H_2CO_3 + 2e^- \rightarrow H_2 + 2HCO_3^- \quad (4)$$

For more details on CO_2 corrosion the reader is referred to a number significant publications covering this field [11–18]. Particular attention is drawn to the recent reviews of the main design considerations [19] and prediction techniques related to CO_2 corrosion [20] compiled by the European Federation of Corrosion.

2.2. Neural networks

Neural networks are models which can be "trained" to forecast, by developing a correlation between a known set of input and output problem descriptors. The learning capability is achieved through its architecture, structured as a network of interconnected non-linear computational units called "neurones", organized in multiple layers, not unlike biological neural cells. Each neurone is characterized by a transfer function, responsible for non-linear interpretation ("learning") of input signals, scaled between 0 and 1. Each connection between the neurones is given a numerical weight, which represents a conductivity of the connection. When the neural network is "learning", the weights are iteratively recalculated until the error of prediction is minimized.

A back propagation neural network [21], used in the present model, consists of three interconnected neurone layers: input, hidden and output layers, as shown in Fig. 1. The network learns by adjusting the connection weights using a gradient-descent technique based on minimization of squared network errors, obtained in

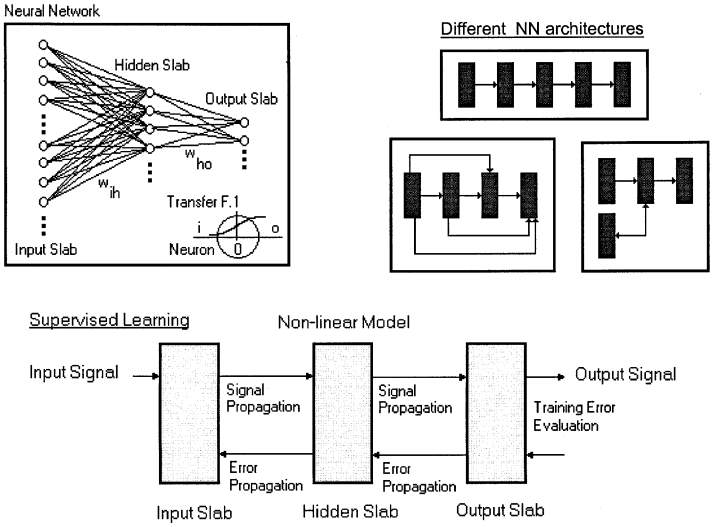

Fig. 1. Back propagation neural network.

comparison between network interpretation of training input patterns and known training outputs. During the training, the network adjusts its weights to obtain better answers simultaneously with backward error propagation through the network layers. The training efficiency depends on interaction between neurone transfer functions and typical training patterns. For more details on the basic theory and application of neural networks see some of the cited references [22,23].

2.3. Monte Carlo method

To account for the uncertainties in the input parameters influencing the output, in this case the corrosion rate, a Monte Carlo method has been used [24]. It is assumed that the one or more of the input parameters have a Gaussian distribution (other distributions can be assumed instead). The Gaussian distribution is specified by giving the mean, μ and the standard deviation, σ (where $-\infty < \mu < \infty$ and $\sigma > 0$). A continuous random variable x is said to have a normal (Gaussian) distribution if the probability density function of x is:

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

Once the mean and the standard deviation for the distribution are given (specified by a user) the question remains how to generate a random value of x which satisfies this distribution. In the present work, this is done by applying the polar form of the Box–Mueller transformation, which generates normally distributed random numbers from uniformly distributed random numbers [25]. The latter can be obtained from numerous random number generators built into most computers. Before each uniform random number is generated a *randomize* function is used to generate a seed for the subsequent random number generation. This is an important step needed to obtain independent random numbers. In this work default random number and randomize generators in Microsoft Excel were used.

The algorithm starts by generating two independent random numbers u_1 and u_2 which come from a uniform distribution (defined between 0 and 1). These numbers are then converted into two independent Gaussian random numbers, g_1 and g_2, with a zero mean and a standard deviation of one. The final Gaussian random variables x_1 and x_2 used as input into the Monte Carlo model are obtained by multiplying g_1 and g_2 with the standard deviation and adding the mean.

$$x_1 = g_1\sigma + \mu$$

$$x_2 = g_2\sigma + \mu$$

For N normally distributed random input parameters, $x_1, x_2, x_3, \ldots, x_N$ the output (corrosion rate) is calculated N times resulting in a distribution of output values $\mathrm{CR}_1, \mathrm{CR}_2, \mathrm{CR}_3, \ldots, \mathrm{CR}_N$. A mean and standard deviation can be calculated as:

$$\overline{\mathrm{CR}} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{CR}_i$$

$$\sigma_{CR} = \left(\frac{1}{N-1} \sum_{i=1}^{N} (CR_i - \overline{CR})^2 \right)^{1/2}$$

3. Neural network model of corrosion

The neural network model presented in this study was created using the Neuro Shell® 2 package [26]. This is a general-purpose software package, which contains 16 common neural network algorithms. It combines ease of use with lots of control over how the networks are trained. Parameter defaults make it easy to get started and the package provides generous flexibility and control for later experimentation. It uses spreadsheet files, but can import other types of data structures. Runtime facilities include a source code generator, 3D graphics (response surfaces), etc. For more details refer to the original www site [26].

The experimental database of Dugstad et al. [18] was used for training of the neural network (courtesy of Institute for Energy Technology, Norway). It covers a broad range of experimental conditions: temperature $t = 20$–$90°C$, pipe flow velocity $v = 0.1$–13 m/s, pH = 3.5–7, CO_2 partial pressure $p_{CO_2} = 0.3$–26 bar, ferrous ion concentration $Fe^{2+} = 1$–230 ppm. The corrosion rate was measured on flat diagonally mounted St52 steel coupons using a radioactive measuring technique in experiments, which lasted from a few days to a few weeks. Long duration experiments were needed to obtain stable corrosion rates due to growth of iron carbide films. Most of the experiments were conducted under conditions where protective films did not form, however, in some of the high temperature experiments precipitation of iron carbonate scales may have occurred. For more details on the experimental techniques and the results see the original paper [18].

All the networks created and tested had five input parameters: water flow velocity, pH, temperature, partial pressure of CO_2 and concentration of Fe^{2+} ions and one output: the corrosion rate. A number of different neural networks were tested by varying the network architecture and by changing the network parameters such as the number of nodes in each layer, the kind of transfer functions used, the training patterns, etc. The best network was selected as the one that had the highest predictive accuracy when compared to the experimental database.

After many experiments, guided primarily by trial and error, the best neural network was identified. [1] It had a standard three-layer network architecture with the hidden layer divided into three "blocks" of neurones (see Fig. 2). Different transfer functions were used for the neurones in each bock (*tanh*, *logistic*, and *Gaussian* in hidden layers 1, 2, and 3 respectively). The neurones in the input layer of this network used a *linear* transfer function for the five input variables, while those in the

[1] This work was done by a fourth year Mechanical Engineering student without any previous experience in neural network modelling.

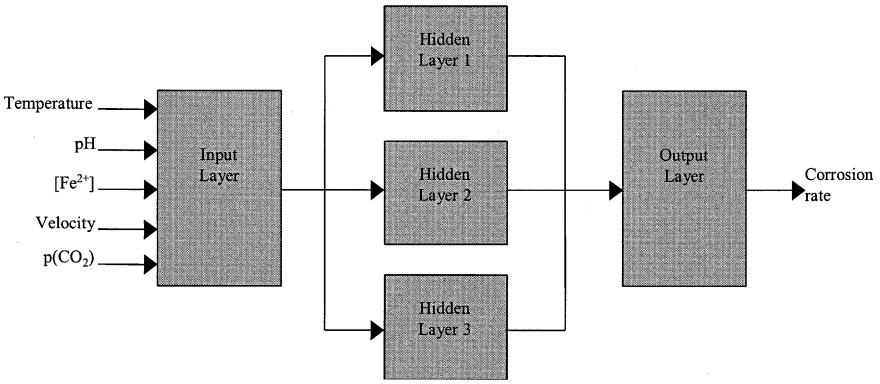

Fig. 2. The optimum neural network architecture identified in the present work.

output layer used a *logistic* transfer function to produce a value for the predicted corrosion rate. This architecture was the most successful one in capturing the complex non-linear relationship between the input and output parameters. The numbers of neurones in each layer were 5 in the input layer, 30 in the hidden layer (10 in each block) and 1 in the output layer. It was found that an increase in the number of neurones in the hidden layer improved the performance of the network up to a point when any further increase did not contribute to the learning process. Actually there is an argument in the literature that when the number of neurones (i.e. connections between the neurones – adjustable weights) exceeds the number of training cases the network begins to *memorise* the correct answers and loses the capability of generalization [22].

3.1. Comparisons

A comparison between the measured corrosion rate values and those predicted by the most successful network is shown in Fig. 3. A good model would have most the points fall close to the diagonal line. The correlation factor for the present model was $r^2 = 0.91$. In order to put the performance of this simple neural network model into a proper perspective the performance of the four previously tested models [1] using the same data is shown below in Fig. 4. It can be seen that the current model performed better than the mechanistic (EC) and the two widely used semi-empirical models (SHELL and IFE) while it was inferior to the more sophisticated neural network model (NN).

3.2. Parametric testing

Parametric tests were performed in order to observe the network's response to variations in input conditions. This was done for two reasons:

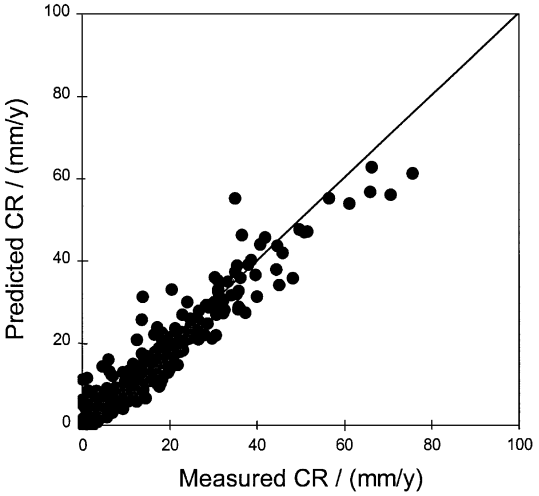

Fig. 3. Comparison between the measured corrosion rates in KSC2 experiments [18] and those predicted by the most successful neural network (shown in Fig. 2).

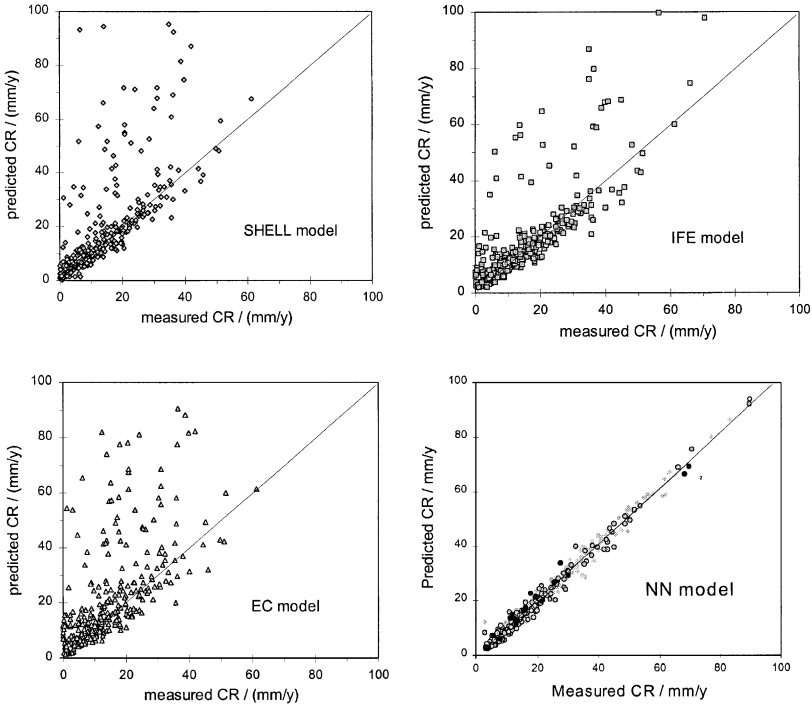

Fig. 4. Performance of the four previously tested models [1] compared with the KSC2 data [18].

(a) non-linear empirical models in general can perform poorly when taken outside the domains where they were trained (calibrated) and sometimes even in between widely spaced experimental values (in the data "gaps");
(b) to test the ability of the developed network to generalize the results and capture the trends suspected to exist in the experimental data.

For these tests, the neural network was given arrays of input data where a single input parameter changed while the others were kept constant. The responses of the neural network are shown below for the cases of varying flow velocity (Fig. 5a), temperature (Fig. 6a), pH (Fig. 7a) and CO_2 partial pressure (Fig. 8a). The sensitivity of the corrosion rate to the changes in Fe^{2+} was negligible and is not shown. Below each of the figures showing the network performance, the corresponding trends extracted from the experimental data are shown. Since in many cases all the exact parameters could not be matched, the closest experimental conditions were selected. In response to the two concerns listed above it can be concluded:

(a) In all cases the predictions are monotonous even for the cases when the experimental points used for training were "wide apart". When taken outside the bounds of the experimental data the network performed also monotonously.
(b) Nearly all of the trends noticed in the experimental data are present in the predictions. For example:

(1) It is known that higher velocities typically lead to higher CO_2 corrosion rates. However, in some instances the dependence can be reversed by the presence of iron carbide films which are known to accelerate corrosion. Their removal at higher velocities can lead to a reduction in the corrosion rate. While this is noticed at lower temperatures, at higher temperatures it is not the case as precipitated iron carbonate might hold the iron carbide in place. This behaviour shown clearly by the experimental results in Fig. 5b is reproduced by the neural network as shown in Fig. 5a.

(2) A general increase in the CO_2 corrosion rate is expected with increasing temperature up to 60–80°C, followed by a reduction at higher temperatures which occurs due to protective iron carbonate film precipitation, as shown in Fig. 6b. This trend is captured by the network (Fig. 6a) and includes the iron carbide related "crossover effect" also seen in the experiments, which was explained in the point above.

(3) The overall reduction of the corrosion rate with increasing pH, seen in the experiments and illustrated in Fig. 7b, is reproduced by the network in Fig. 7a. The "crossover effect" in this case, seen both in the experiments and the predictions, is related to the absence or very loose adherence of iron carbide films at low pH.

(4) Finally, the dependence of the corrosion rate on CO_2 partial pressure is shown in Fig. 8a and b. While the suspected 0.7 power law relationship, suggested by the pioneering work of de Waard and Milliams [12], is difficult to discern in the experimental data, the predictions clearly confirm this trend for $p_{CO_2} > 1$ bar. For CO_2 partial pressures below 1 bar the dependency fades to zero as would be expected from theory.

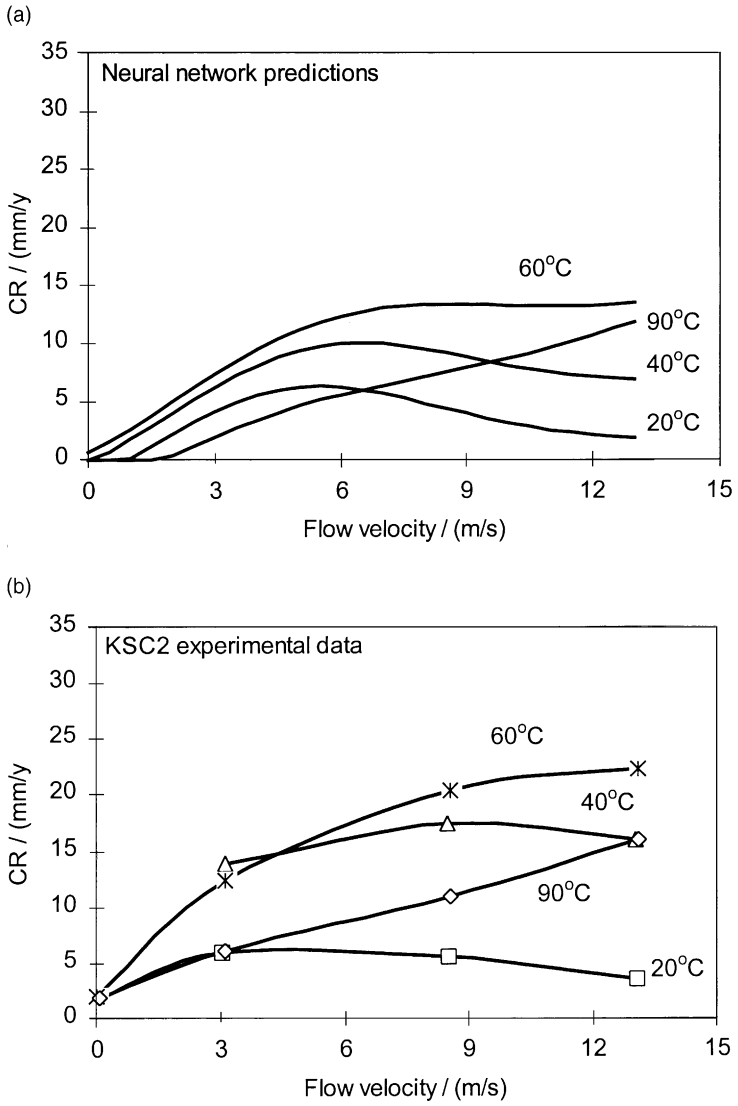

Fig. 5. (a) Predicted sensitivity of the corrosion rate to flow velocities change for a range of temperatures; other conditions: $p_{CO_2} = 2$ bar, pH = 5, $[Fe^{2+}] = 50$ ppm. (b) Measured sensitivity of the corrosion rate to flow velocity change for a range of temperatures; data taken from the KSC2 database [18]; other conditions: unbuffered water, $p_{CO_2} = 2$ bar. Points represent measurements, the lines highlight the trends.

4. Probabilistic predictions

A probabilistic Monte Carlo method was applied to the neural network model of CO_2 corrosion described above by randomizing the input parameters. This was

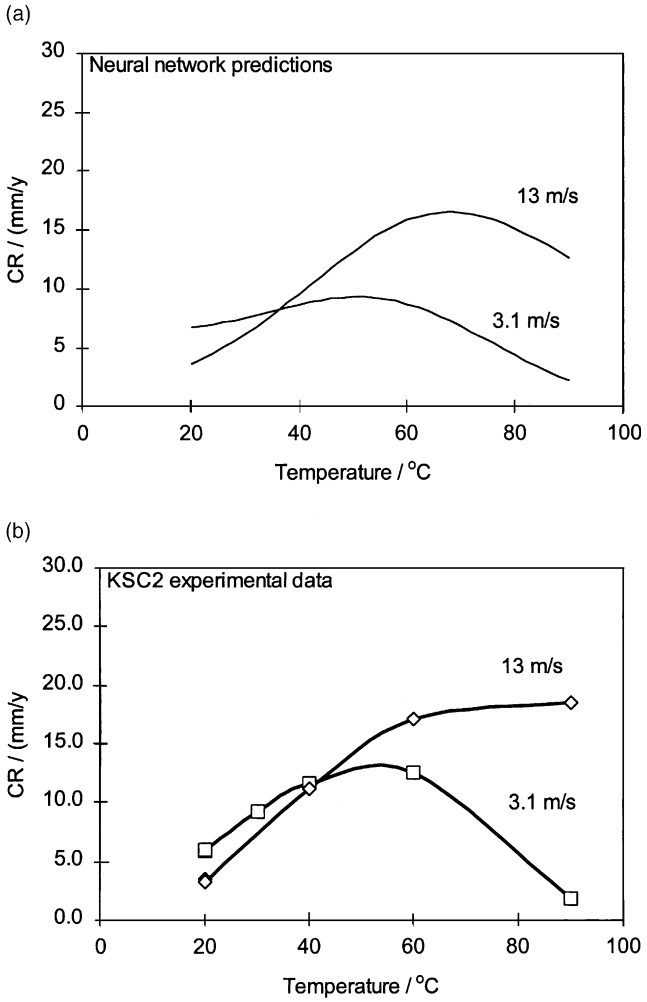

Fig. 6. (a) Predicted sensitivity of the corrosion rate to temperature change for two different flow velocities; other conditions: $p_{CO_2} = 2.5$ bar, pH = 5, $[Fe^{2+}] = 50$ ppm. (b) Measured sensitivity of the corrosion rate to temperature change for two different flow velocities; data taken from the KSC2 database [18]; $p_{CO_2} = 2.5$ bar. Points represent measurements, the lines highlight the trends.

achieved by defining a mean value and a standard deviation for a given parameter and by assuming a normal (Gaussian) distribution. In most cases one parameter was randomized at a time while all other input parameters were kept constant. By changing the mean and the standard deviation, the sensitivity of the model to the selected input parameter could be determined.

Two resulting distributions of the corrosion rate obtained for a normal distribution of input flow velocity are shown in Fig. 9. It is clear that higher flow velocity leads to a higher corrosion rate but also that the corrosion rate is not sensitive to a

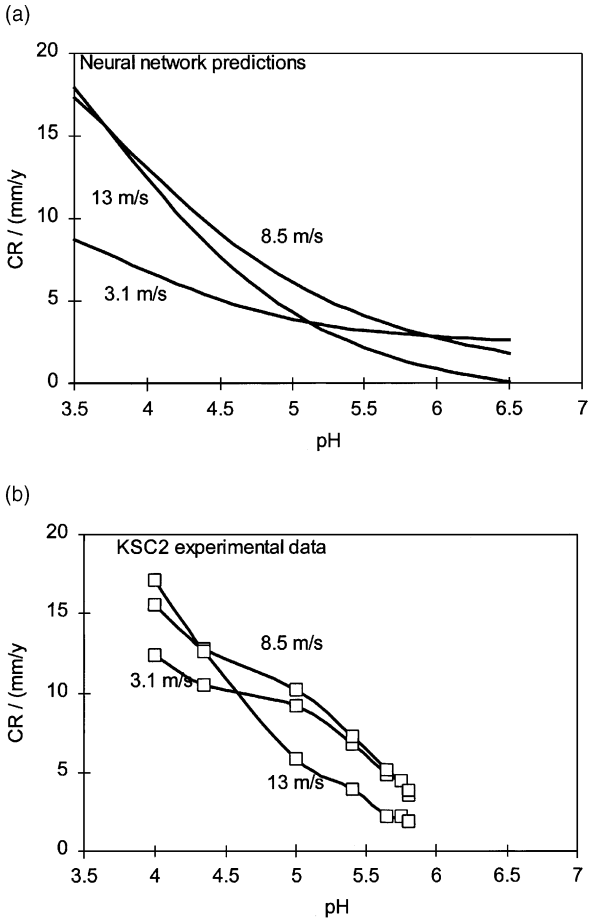

Fig. 7. (a) Predicted sensitivity of the corrosion rate to pH change for three different flow velocities; other conditions: $p_{CO_2} = 1.4$ bar, $T = 40°C$, $[Fe^{2+}] = 50$ ppm. (b) Measured sensitivity of the corrosion rate to pH change for three different flow velocities; data taken from the KSC2 database [18]; other conditions: $p_{CO_2} = 1.4$ bar, $T = 40°C$. Points represent measurements, the lines highlight the trends.

variation in the flow velocity in this regime. The explanation for this can be found by looking at Fig. 9. The same variation in the input flow velocity should have a larger effect on the corrosion rate at lower velocities as it represents a larger relative change. However, the real reason for the observed behaviour can be found by looking at Fig. 5 where the corrosion rate vs. flow velocity curve for 40°C has a near-zero slope at 8 m/s while at 2 m/s the slope is positive. It should also be noted that at the low velocity the resulting distribution of the corrosion rate is near-normal (given the normal distribution of the input flow velocity).

A similar conclusion can be drawn from Fig. 10 where the sensitivity of the corrosion rate to a variation in pH is shown. Without exception, a narrow distri-

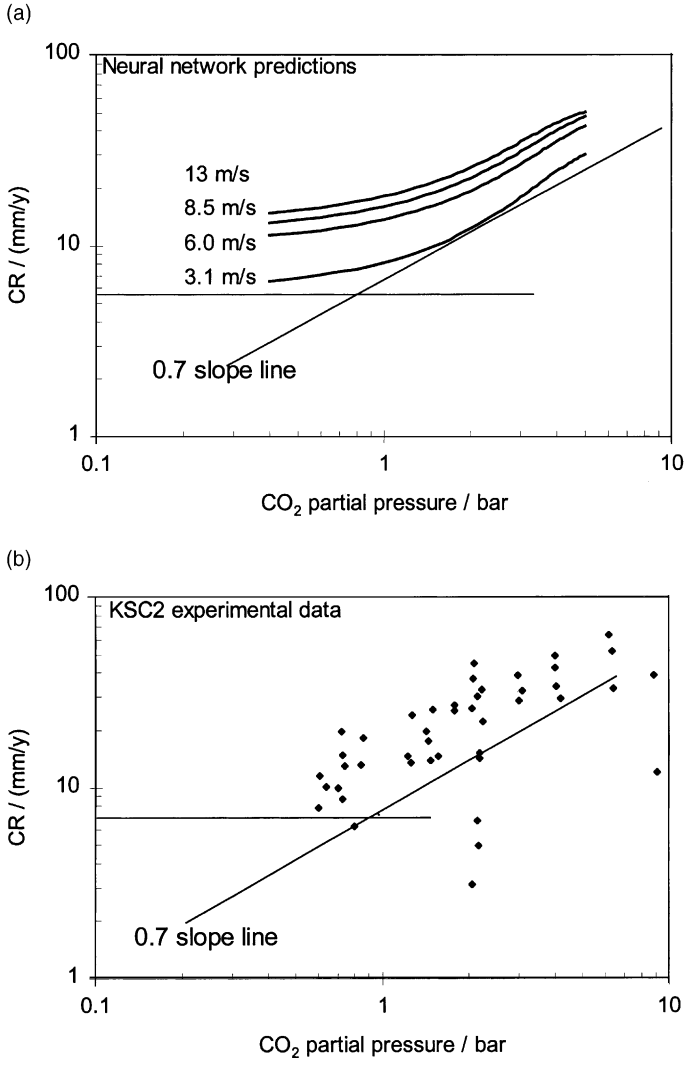

Fig. 8. (a) Predicted sensitivity of the corrosion rate to CO_2 partial pressure change for different flow velocities; other conditions: $T = 60°C$, $[Fe^{2+}] = 50$ ppm. The lines indicate zero and 0.7 power law dependency. (b) Measured sensitivity of the corrosion rate to CO_2 partial pressure change for different flow velocities; data taken from the KSC2 database [18]; $T = 60°C$. Points represent measurements, the lines indicate zero and 0.7 power law dependency.

bution of corrosion rates is predicted at pH 6.5 while at pH 4 a broad near-normal distribution is obtained at much higher values. For interpretation use Fig. 7. Yet another similar example is shown in Fig. 11 for p_{CO_2} sensitivity with reference to Fig. 8. Note that the narrowest corrosion rate distribution is obtained for the case with the largest relative change in p_{CO_2}.

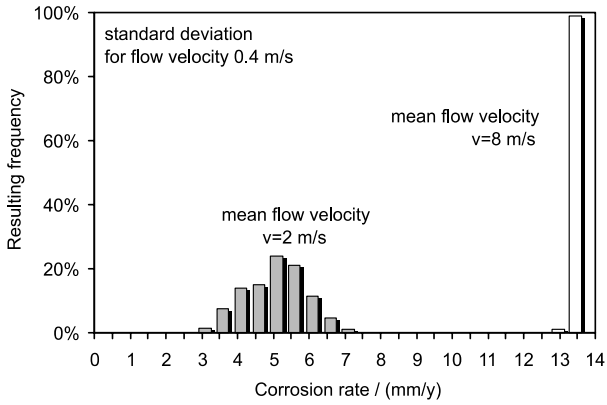

Fig. 9. Distribution of the corrosion rate resulting from a normal variation in flow velocity (other conditions: $T = 40°C$, $[Fe^{2+}] = 50$ ppm, pH = 4, $p_{CO_2} = 1.4$ bar).

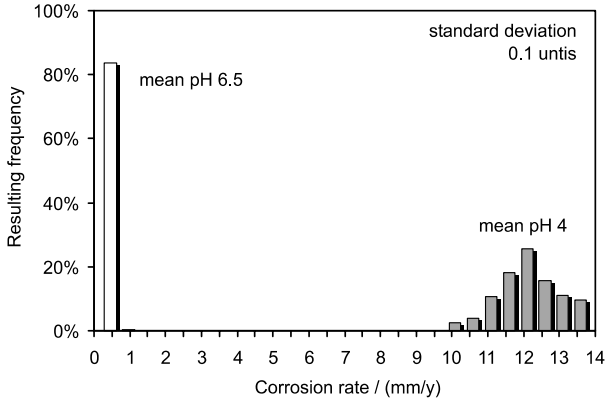

Fig. 10. Distribution of the corrosion rate resulting from a normal variation in pH (other conditions: $T = 40°C$, $[Fe^{2+}] = 50$ ppm, $v = 13$ m/s, $p_{CO_2} = 1.4$ bar).

Interesting behaviour can be observed in Fig. 12. The two selected temperatures are not far apart however two very different variations in the corrosion rate are obtained – a near-normal distribution for 50°C and a very skewed distribution for 70°C. The explanation can once again be found by looking back at Fig. 6. Temperature of 50°C relates to the sloped part of the curve suggesting that a variation of temperature around 50°C would result in corrosion rates, either larger or smaller than the ones obtained for 50°C. However, 70°C is very near the maximum point in Fig. 6. Therefore any variation of temperature around 70°C (be it positive or negative) always results in a smaller prediction for the corrosion rate than obtained at 70°C, hence a skewed distribution in Fig. 12.

Another way of using the probabilistic predictions is to maintain the mean value of the input parameters constant while varying the standard deviation. One example

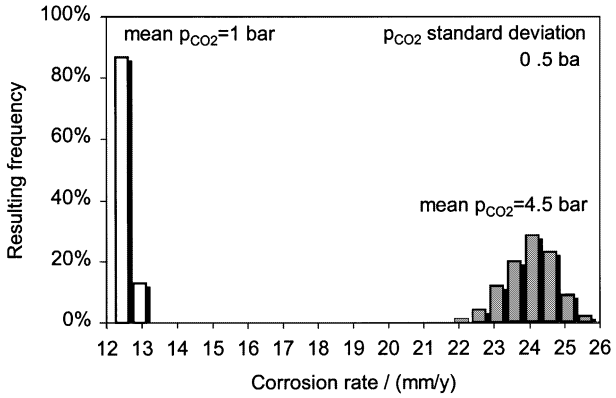

Fig. 11. Distribution of the corrosion rate resulting from a normal variation in p_{CO_2} (other conditions: $T = 80°C$, $[Fe^{2+}] = 50$ ppm, $v = 13$ m/s, pH = 5).

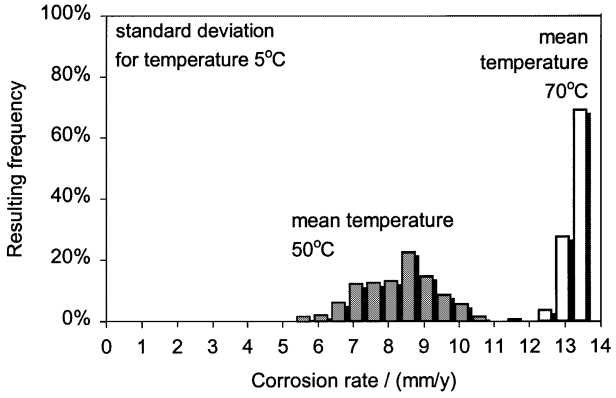

Fig. 12. Distribution of the corrosion rate resulting from a normal variation in temperature (other conditions: pH = 5, $[Fe^{2+}] = 50$ ppm, $v = 13$ m/s, $p_{CO_2} = 1.4$ bar).

is shown in Fig. 13. There a mean temperature of 55°C was used with two different standard deviations of 1°C and 6°C. In the results it can be observed that for a standard deviation of 1°C the resulting corrosion rate distribution had a mean value of 9.8 mm/year with a standard deviation of 0.36 mm/year. This practically means that nearly all the predictions were between 9 and 11 mm/year. For a standard deviation of 6°C, the corrosion rate distribution had a mean value of 10 mm/year and a standard deviation of 1.9 mm/year meaning that the results were scattered between 2.5 and 14 mm/year. Practical implications of these results are straightforward. When using the above neural network package to predict the CO_2 corrosion rate, if one operates with an uncertainty in temperature of 1°C there should be little concern with respect to the resulting uncertainty in output results. On the other hand, an uncertainty in temperature of 6°C would mean an unacceptable scatter in the

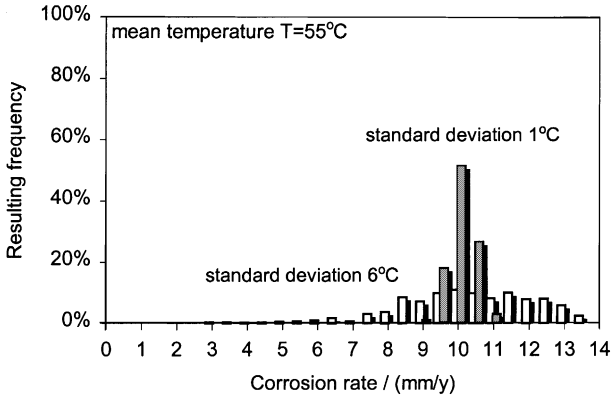

Fig. 13. Distribution of the corrosion rate resulting from a normal variation in temperature (other conditions: pH = 5, $[Fe^{2+}] = 50$ ppm, $v = 13$ m/s, $p_{CO_2} = 1.4$ bar).

predictions. Similar arguments could be drawn about the uncertainties with respect to velocity, pH and p_{CO_2} shown in Figs. 14–16.

So far results were presented where only one parameter at a time was randomized while other parameters were held constant. For a careful analyst, although convenient this technique did not offer much new information compared to the parametric study shown above. However, when there exist simultaneous uncertainties in two or more parameters, a deterministic analysis using the parameter study plots becomes rather complicated. On the other hand, the probabilistic Monte Carlo method works just as smoothly with any number of randomized parameters. The only price to be paid is an increasing sample size required for getting reproducible distributions.

To illustrate this point, a base case was selected: temperature $t = 50°C$, pipe flow velocity $v = 8$ m/s, pH 4, CO_2 partial pressure $p_{CO_2} = 1.4$ bar, ferrous ion concentration $Fe^{2+} = 50$ ppm. Given these inputs the deterministic prediction using the

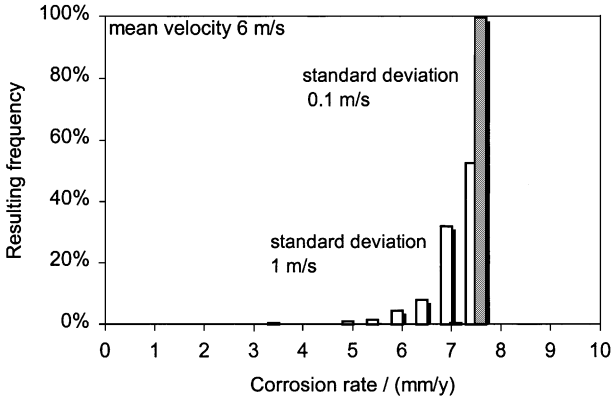

Fig. 14. Distribution of the corrosion rate resulting from a normal variation in velocity (other conditions: pH = 5, $[Fe^{2+}] = 50$ ppm, $T = 40°C$, $p_{CO_2} = 1.4$ bar).

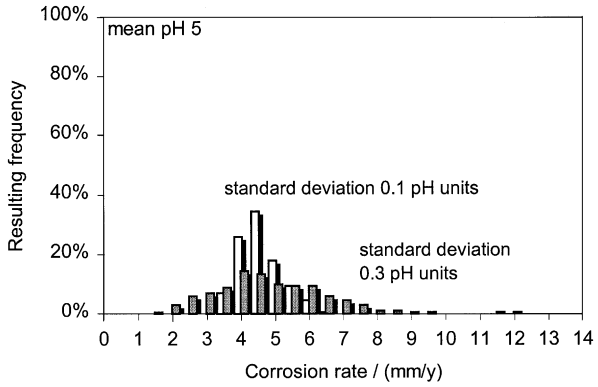

Fig. 15. Distribution of the corrosion rate resulting from a normal variation in pH (other conditions: $T = 40°C$, $[Fe^{+}] = 50$ ppm, v = 13 m/s, $p_{CO_2} = 1.4$ bar).

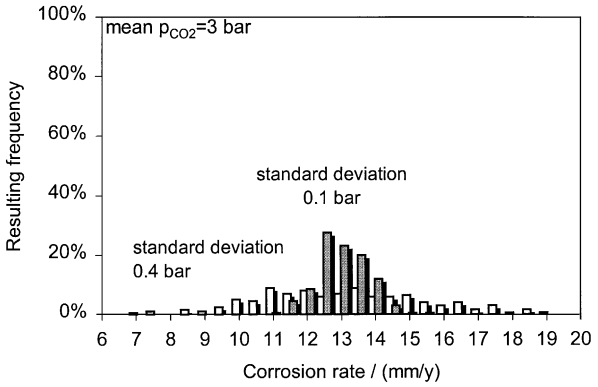

Fig. 16. Distribution of the corrosion rate resulting from a normal variation in p_{CO_2} (other conditions: pH = 5, $[Fe^{2+}] = 50$ ppm, $v = 13$ m/s, $T = 40°C$).

neural network is a corrosion rate of 15.88 mm/year. Subsequently, velocity, pH and temperature were randomized. The standard deviations used were for velocity: $\sigma_v = 0.4$ m/s, for pH: $\sigma_{pH} = 0.1$ pH units and for temperature: $\sigma_t = 5°C$. The resulting corrosion rate distributions in case of individual variations of the three parameters is shown in Fig. 17. The mean and standard variations of the predicted corrosion rate are listed in Table 1 below. It is clear that the mean predicted corrosion rate is similar in all cases while the biggest scatter in the predictions is obtained due to individual temperature variation. The standard deviation obtained with individual temperature variation is nearly an order of magnitude higher than the one obtained for velocity variation. When more than one input parameter is varied simultaneously, the mean does not change significantly. The standard deviation is just slightly higher than the larger of the two obtained for individual variation of the same parameters. The same conclusion applies when all three parameters

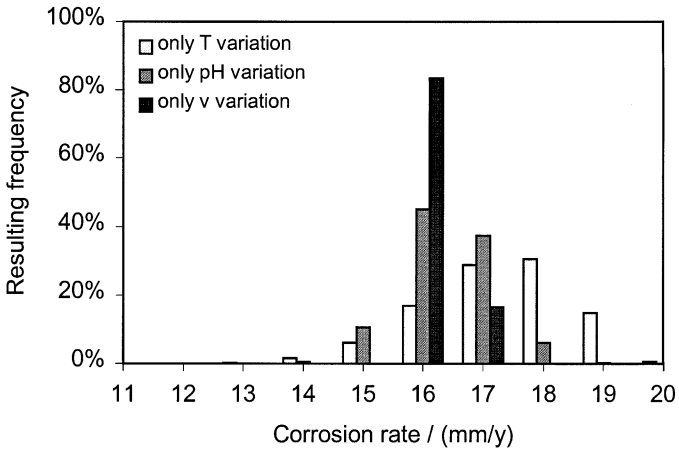

Fig. 17. Distribution of the corrosion rate resulting form individual variations of velocity, pH and temperature. Mean input parameters: pH = 4, $[Fe^{2+}] = 50$ ppm, $v = 8$ m/s, $T = 50°C$, $p_{CO_2} = 1.4$ bar. Standard deviations $\sigma_v = 0.4$ m/s, $\sigma_{pH} = 0.1$ pH units, $\sigma_t = 5°C$.

are varied simultaneously as illustrated in Fig. 18, which is remarkably similar to Fig. 17. A practical consequence for a corrosion engineer should be comforting: it is sufficient to identify the parameter producing the largest uncertainty, while the others then contribute very little to the overall variability of the predictions.

5. Conclusions

(1) Based on experimental data, an effective neural network model of CO_2 corrosion was created using a standard commercial general-purpose software package and simple modelling strategies. The developed model performed better then the

Table 1
Mean and standard deviation of the predicted corrosion rate resulting from variations of velocity, pH and temperature[a]

	Mean predicted corrosion rate (mm/year)	Standard deviation in the predicted corrosion rate (mm/year)
Only velocity variation	15.85	0.16
Only pH variation	15.89	0.73
Only temperature variation	15.77	1.18
Velocity and pH variation	15.85	0.74
Velocity and temperature variation	15.75	1.20
pH and temperature variation	15.79	1.36
Velocity, pH and temperature variation	15.85	1.39

[a] Mean input parameters: $v = 8$ m/s, pH = 4, $T = 50°C$, $[Fe^{2+}] = 50$ ppm, $p_{CO_2} = 1.4$ bar. Standard deviations $\sigma_v = 0.4$ m/s, $\sigma_{pH} = 0.1$ pH units, $\sigma_t = 5°C$.

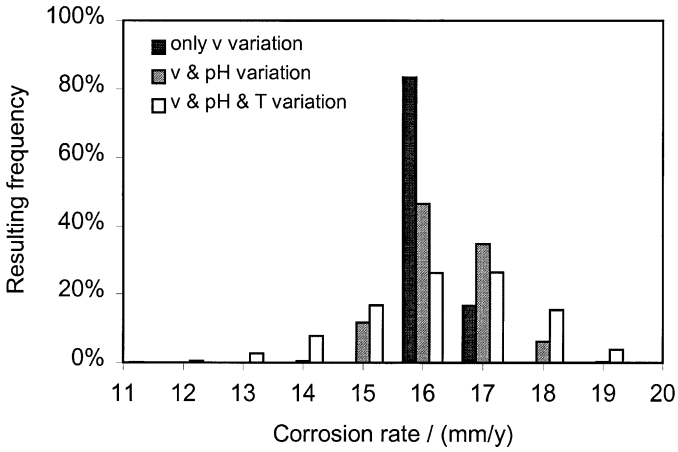

Fig. 18. Distribution of the corrosion rate resulting form variations of velocity, pH and temperature. Mean input parameters: pH = 4, $[Fe^{2+}] = 50$ ppm, $v = 8$ m/s, $T = 50$°C, $p_{CO_2} = 1.4$ bar. Standard deviations $\sigma_v = 0.4$ m/s, $\sigma_{pH} = 0.1$ pH units, $\sigma_t = 5$°C.

previously tested mechanistic and two widely used semi-empirical models, while it was inferior to the more sophisticated neural network model, all tested with the same data.

(2) Parametric testing of the model was performed in order to observe the network's response to variations in input conditions. In all cases the predictions were monotonous even if the experimental points used for training were in some cases "wide apart". When taken outside the bounds of the experimental data, the network performed monotonously. Nearly all of the trends noticed in the experimental data were captured by the network.

(3) A neural network model such as the one described above could be readily developed by a corrosion engineer for any problem at hand, given that sufficient experimental data exist. This applies even in the cases when the understanding of the underlying processes is poor.

(4) A probabilistic Monte Carlo method was implemented and linked to the neural network model of CO_2 corrosion. The combined model was tested by randomizing the input parameters. A number of comparisons presented above have illustrated how a corrosion engineering might use this approach to rapidly test the sensitivity of the model to the uncertainties associated with the input parameters.

Acknowledgements

The experimental database and partial financial support for this project was provided by the Institute for Energy Technology in Norway. Further support has been obtained through a grant from the Australian Research Council. Both contributions are greatly appreciated.

References

[1] S. Nesic, J. Postlethwaite, M. Vrhovac, CO_2 corrosion of carbon steel – from mechanistic to empirical modelling, a review article, J. Corros. Rev. 15 (1997) 211.
[2] A. Anderko, R. Young, Simulation of CO_2/H_2S corrosion using thermodynamic and electrochemical models, Corrosion/99, Paper no. 31 Houston, TX, NACE International, 1999.
[3] E. Dayalan, F.D. de Moraes, J.R. Shadley, S.A. Shirazi, E.F. Ribicki, CO_2 corrosion prediction in pipe flow under $FeCO_3$ scale-forming conditions, Corrosion/98, Paper no. 51, Houston, TX, NACE International, 1998.
[4] R. Zhang, M. Gopal, W.P. Jepson, Development of a mechanistic model for predicting corrosion rate in multi-phase oil/water/gas flows, Corrosion/97, Paper no. 601, Houston, TX, NACE International, 1997.
[5] M. Sundaram, V. Raman, M.S. High, D.A. Tree, J.Wagner, Deterministic modeling of corrosion in downhole environments, Corrosion/96, Paper no. 30, Houston, TX: NACE International, 1996.
[6] S. Nesic, M. Vrhovac, A neural model for prediction of CO_2 corrosion of steel, J. Corros. Sci. Engng. ISSN 1466-8858, http://www.cp.umist.ac.uk/ JCSE/vol1/paper6/vl_p6. html, 1999.
[7] H.M.G. Smets, W.F.L. Bogaerts, Corrosion 48 (1992) 618.
[8] S.P. Trassati, F. Mazza, Br Corros. J. 31 (1996) 105.
[9] E.M. Rosen, D.L. Silverman, Corrosion 48 (1992) 734.
[10] J. Cai, R.A. Cottis, S.B. Lyon, Corros. Sci. 41 (1999) 2001.
[11] C. de Waard, D.E. Milliams, Predication of carbonic acid corrosion in natural gas pipelines, First International Conference on the Internal and External Corrosion of Pipes, Paper F1, University of Durham, England, 1975.
[12] C. de Waard, D.E. Milliams, Corrosion 31 (1975) 131.
[13] G. Schmitt, B. Rothman, Werkstoffe und Korrosion 28 (1977) 816.
[14] L.G.S. Gray, B.G. Anderson, M.J. Danysh, P.R. Tremaine, Mechanism of carbon steel corrosion in brines containing dissolved carbon dioxide at pH 4. Corrosion/89, Paper no. 464, Houston, TX, NACE International, 1989.
[15] L.G.S. Gray, B.G. Anderson, M.J. Danysh, P.R. Tremaine, Effect of pH and temperature on the mechanism of carbon steel corrosion by aqueous carbon dioxide, Corrosion/90, Paper no. 40, Houston, TX, NACE International, 1990.
[16] C. de Waard, U. Lotz, Prediction of CO_2 corrosion of carbon steel, Corrosion/93, Paper no. 69, Houston, TX, NACE International, 1993.
[17] S. Nesic, J. Postlethwaite, S. Olsen, An electrochemical model for prediction of CO_2 corrosion, Corrosion/95, Paper no. 131, Houston, TX, NACE International, 1995.
[18] A. Dugstad, L. Lunde, K. Videm, Parametric study of CO_2 corrosion of carbon steel, Corrosion/94, Paper no. 14, Houston, TX, NACE International, 1994.
[19] CO_2 corrosion control in oil and gas production, A Working Party Report, European Federation of Corrosion Publication 23, The Institute of Materials, London, England, 1997.
[20] Predicting CO_2 Corrosion in the oil and gas industry, A Working Party Report, European Federation of Corrosion Publication 13, The Institute of Materials, London, England, 1994.
[21] W.C. Carpenter, M.E. Hofman, Training backprop neural networks, Journal AI Expert, A Miller Freeman Publication, USA, 1995, pp. 30–33.
[22] P. Simpson, Artificial Neural Systems. Pergamon Press, New York, 1990.
[23] P. Wasserman, Neural Computing, Theory and Practice, Van Nostrand Reinhold, New York, 1989.
[24] I.M. Sobol, A Primer for the Monte Carlo Method, CRC Press, Florida, 1994.
[25] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes – the Art of Scientific Computing, Cambridge University Press, New York, 1986.
[26] http://www.wardsystems.com/.