# Some thoughts on neural network modelling of microabrasion–corrosion processes

P. Srinivasa Pai[a,*], M.T. Mathew[b], M.M. Stack[c], L.A. Rocha[b,d]

[a]*Department of Mechanical Engineering, NMAM Institute of Technology, Nitte 574110, Karnataka, India*
[b]*Research Centre on Interfaces and Surfaces Performance, Azurém, 4800-058 Guimarães, Portugal*
[c]*Department of Mechanical Engineering, University of Strathclyde, James Weir Building, 75 Montrose Street, Glasgow G1 2XJ, UK*
[d]*Universidade do Minho, Department of Engineering Mecânica, Azurém, 4800-058 Guimarães, Portugal*

## Abstract

There is increasing interest in the interactions of microabrasion, involving small particles of less than 10 μm in size, with corrosion. This is because such interactions occur in many environments ranging from the offshore to health care sectors. In particular, microabrasion–corrosion can occur in oral processing, where the abrasive components of food interacting with the acidic environment, can lead to degradation of the surface dentine of teeth.

Artificial neural networks (ANNs) are computing mechanisms based on the biological brain. They are very effective in various areas such as modelling, classification and pattern recognition. They have been successfully applied in almost all areas of engineering and many practical industrial applications.

Hence, in this paper an attempt has been made to model the data obtained in microabrasion–corrosion experiments on polymer/steel couple and a ceramic/lasercarb coating couple using ANN. A multilayer perceptron (MLP) neural network is applied and the results obtained from modelling the tribocorrosion processes will be compared with those obtained from a relatively new class of neural networks namely resource allocation network.

© 2007 Elsevier Ltd. All rights reserved.

*Keyword:* Microabrasion–corrosion process; Artificial neural network (ANN); Multilayer perceptron (MLP); Radial basis function (RBF); Resource allocation network

## 1. Introduction

Tribocorrosion involving the interactions of tribological phenomena, such as sliding wear, abrasion or erosion, and chemical phenomena, such as dissolution or passivation, is a complex process and there are many different possible physical interactions, where each of these processes are modified either in a positive or negative direction [1]. Many attempts have been made to investigate the tribocorrosion behavior of various metallic couples using traditional tribometers, such as the reciprocating sliding tribometer and rotating disk apparatus [2,3]. Although there have been a number of studies carried out on the microabrasion

performance of materials and involved mechanisms in the process [4–9], few of these have investigated the individual components of the tribocorrosion interaction at the contact using electrochemical methods. There have, however, been a number of recent attempts to study the effects of microabrasion on the corrosion rate in a quantitative manner [10,11]. These effects are important to characterize, as tribological processes in bio-medical conditions typically occur in corrosion solutions in which the pH and electrochemical potential may vary significantly [12,13]. Understanding the combined effects of the tribological and corrosion variables is vitally important for optimizing materials selection and tribocorrosion parameters in such conditions.

Artificial neural networks (ANNs) have been used to model tribological processes and have provided promising

---

*Corresponding author.
E-mail address:* mathew@dem.uminho.pt (P. Srinivasa Pai).

**Nomenclature**

| | |
|---|---|
| $\kappa$ | overlap parameter |
| $\delta$ | scale of resolution in the input space |
| $\xi_i$ | input to a neuron |
| $\sigma_j$ | width of $j$th RBF unit |
| $\zeta_k$ | output to a neuron |
| $\Delta K_a$ | the effect of corrosion on the microabrasion |
| $\Delta K_c$ | the effect of microabrasion on corrosion |
| $\delta_{max}$ | maximum distance above which RBF units are added |
| $\delta_{min}$ | minimum distance |
| ANN | artificial neural network |
| $b$ | crater diameter |
| BP | back-propagation |
| $d$ | distance between center and input pattern |
| $e_{min}$ | desired approximation accuracy of the network output |
| $K_a$ | microabrasion weight loss |
| $K_{ac}$ | microabrasion–corrosion weight loss |
| $K_{ao}$ | microabrasion weight loss in the absence of corrosion |
| $K_c$ | corrosion weight loss |
| $K_{co}$ | the corrosion weight loss in the absence of wear |
| MLP | multilayer perceptron |
| $O_k$ | overall network output of neuron $k$ |
| $R$ | ball radius |
| RBF | radial basis function |
| $V$ | wear volume |
| $v_j$ | net internal activity of neuron $j$ |
| $w_{ji}$ | weights between neuron $j$ in the hidden layer and neuron $i$ in the input layer |
| $w_{kj}$ | weights connecting hidden neuron $j$ to output neuron $k$ |
| $x_j$ | center of $j$th RBF unit |
| $y_j$ | the output of a neuron |
| $\alpha$ | momentum term |
| $\eta$ | learning rate |
| $\tau$ | decay constant |

results [14]. The main functions performed by ANNs were predictions (model) and classifications of the process. Prediction may be used for diagnosis, accelerated life-time testing, on-line control of manufacturing processes that involve wear and prediction of the main properties of the mechanical systems, during the conceptual design stage. Both supervised and unsupervised models were used successfully to model the process.

Hence, the main objective of the current work is to employ neural networks to model the obtained results from the microabrasion–corrosion experiments, an area not tackled to date by ANN modelling approaches. Experimental data have been collected from microabrasion–corrosion tests on two couples, involving polymer-steel and a ceramic–lasercarb coating. These data have been used to train and test a MLP neural network, which is commonly and widely used network architecture. The objective of modelling using a neural network has been to estimate $K_{ac}$ and $K_a$ namely the total microabrasion–corrosion rate and the total microabrasion rate, respectively. These results have been compared with that obtained from another network namely resource allocation network (RAN), a radial basis function based neural network. Based on the findings, the relative advantages and disadvantages of both modelling approaches are discussed in the paper.

## 2. Data collection and modelling

### 2.1. Some aspects on experimental details

Microabrasion tests were performed on a commercially available apparatus, the TE-66, microabrasion tester, Fig. 1, supplied by the Phoenix Tribology Ltd, UK (formerly known as Plint and Partners Ltd, UK).

Essentially in the microabrasion tester, a 25 mm ball is located between two-coaxial shafts, each carried in a support bearing. One shaft was driven by a variable speed DC geared motor. The test sample was clamped onto a platform, which was fitted to the pivoted L-shaped arm. The corrosive slurry was stored in a container that could be agitated by means of a laboratory magnetic stirrer and was delivered to the specimen by an integral peristaltic pump. The slurry (concentration of $0.025\,\mathrm{g\,cm^{-3}}$) was fed to a position just above the contact point and collected in a waste tray underneath. The arm, which holds the sample, could be moved horizontally in order that several tests on a single sample specimen could be carried out. Following the test, the worn samples were examined by optical, scanning electron and atomic force microscopy. The approximate error in the experimental data was estimated to be $\pm 20\%$.

The wear volume was calculated using the standard technique for measuring the wear scar of spherical geometry [5] i.e. the geometry of the wear scar is assumed to reproduce the spherical geometry of the ball, and the wear volume '$V$' may then be given by

$$V \approx \pi b^4 / 64R \quad \text{for } b \ll R, \tag{1}$$

where $R$ is the ball radius and $b$ is crater diameter.

For estimating the corrosion rate, the sample was connected to the working electrode and a reference electrode was connected by capillary tube in order to make contact with the circuit. A Pt–Ti wire mesh was used as an auxiliary electrode. Potential control for corrosion studies was carried out using a Gill AC electrochemical interface (ACM Instruments, UK). The present study is focused on data modelling; hence, other particulars on the experimental procedure are illustrated in the published work [11]. The microabrasion–corrosion data were used from two
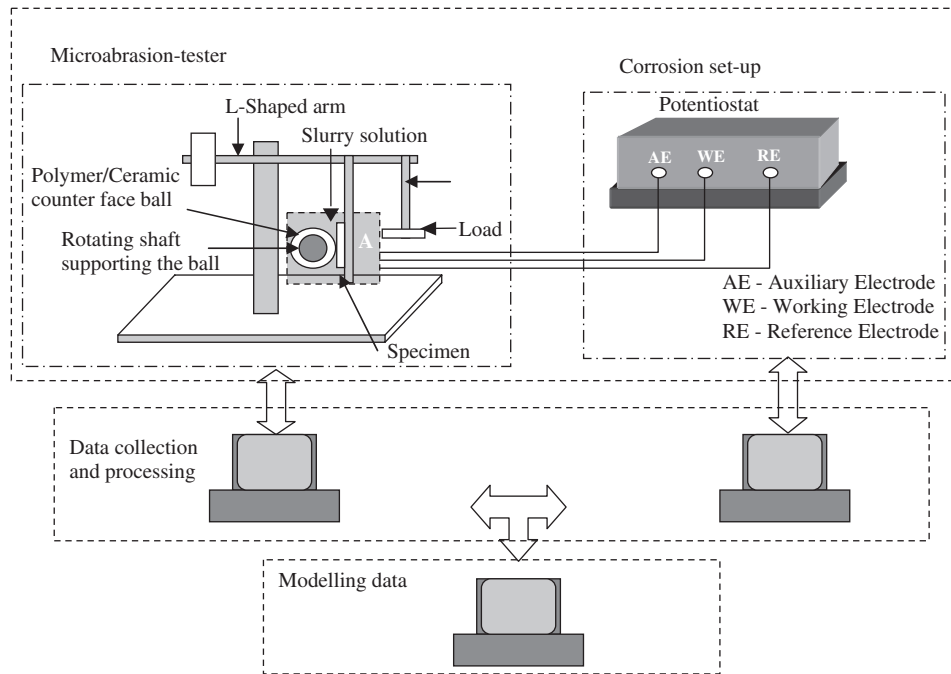
Fig. 1. Schematic diagram of the experimental setup.

Table 1
Microabrasion–corrosion experimental details for polymer/mild steel couple

| | |
|---|---|
| Sample materials | Mild steel |
| Ball materials | Polypropylene |
| Speed | 100 rpm |
| Load | 1–5 N |
| Sliding distance | 3000 rev or (235.50 m) |
| Potentials | −0.6, −0.4, −0.2, 0, +0.2 V (E vs. SCE) |
| Slurry | Slurry composition: silicon carbide (SiC, 4 μm diameter) with corrosive liquid ($0.5Na_2CO_3 + 0.5NaHCO_3$) (concentration of 0.025 g cm$^{-3}$) |

Table 2
Microabrasion–corrosion experimental details for ceramic/lasercarb coating couple

| | |
|---|---|
| Sample materials | Ni–Cr/WC lasercarb coating (composition: C—17.92%, Cr—9.33%, Ni—59.24%, W—7.39%, Si—4.76% Fe—1.35%, thickness: 1189 μm, hardness: 689 VHN) |
| Ball materials | Ceramic ($Si_3N_4$) |
| Speed | 100 rpm |
| Load | 1–5 N |
| Sliding distance | 3000 rev or (235.50 m) |
| pHs | 2, 5.5, 9.7, 13.5 |
| Slurry | Slurry composition: silicon carbide (4 μm diameter) with corrosive liquid (concentration of 0.025 g cm$^{-3}$). The chemical composition of the solutions are as follows |
| | pH  Composition of solution |
| | 2  Anhydrous sodium sulphate + sodium hydrogen sulphate monohydrate (0.5 M) |
| | 5.5  Anhydrous sodium dihydrogen phosphate + anhydrous disodium hydrogen phosphate (0.5 M) |
| | 9.7  Sodium bicarbonate + Sodium carbonate (0.5 M) |
| | 13.5  Potassium hydroxide (0.5 M) |

couples namely (a) Polymer–steel couple and (b) Ceramic-lasercarb coating couple. More details on the experimental work are given in Tables 1 and 2. The SEM image of the SiC abrasive particles and an optical image of the cross-section of the lasercarb coating are shown in Figs. 2(a) and (b), respectively. Lasercarb coating is a special type of coating, having a thickness of 1189 μm, supplied by Weir Pumps Ltd., Glasgow, UK. The principle involves using the energy of a $CO_2$ laser to lightly melt the filler metal, such as carbide powders and the base metal. The lasercarb process does not affect carbide grains, which maintain their intrinsic properties. The deposits are metallurgically welded to the base metal. They are also extremely dense, very precise and easily reproducible. Lasercarb coatings are used successfully in various industries, such as oil drilling, steel, papermaking, food processing and waste recycling. Typical SEM images of the worn surfaces are shown in Figs. 2(c) and (d).

As it is shown in Tables 1 and 2, in the case of polymer/mild steel couple, the variables are load and applied potential and for ceramic/lasercarb coating couple, load and solutions of different pH were used. Hence, load, applied potential and pH of the solutions were input parameters to the NN model and output is the micro-abrasion–corrosion ($K_{ac}$) and corrosion weight losses ($K_c$), as shown in the schematic diagram (Fig. 3).
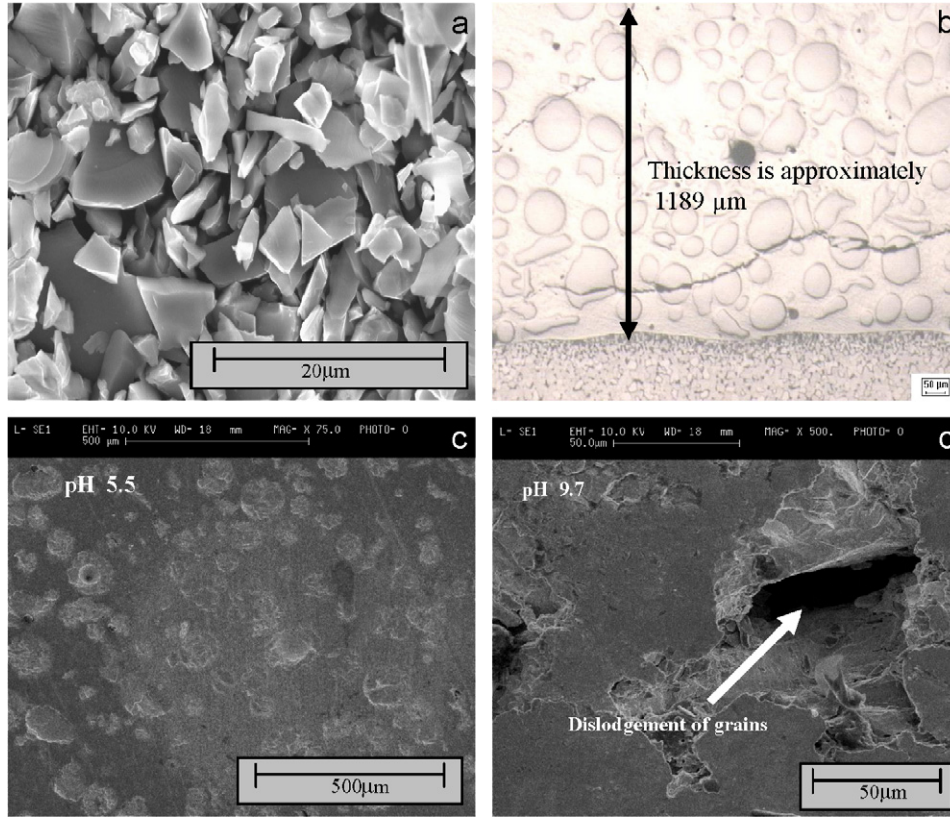
Fig. 2. (a)–(d) Optical and SEM images, (a) SiC-abrasive particles used in the test, (b) cross-sectional view of the lasercarb coating, (c) wear scar, lasercarb coating, 5N, (d) worn surface, lasercarb coating, 5N, high magnification.
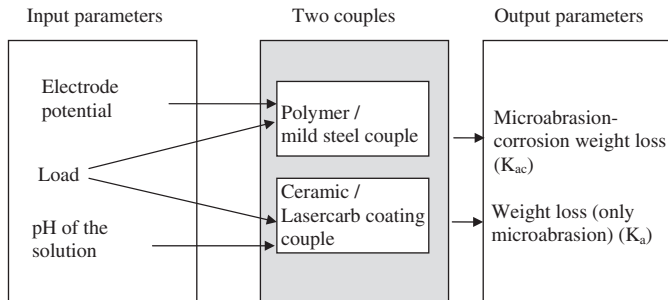


Fig. 3. Schematc diagram of the structure of the NN model, showing the input parameters and output/results of the model.

## 2.2. Modelling using ANN

ANNs are computational systems that simulate the neurons of a biological nervous system. Basically, all ANNs have a similar topological structure or architecture [15]. Generally there are three layers—an input layer, which receives information from the external world, a hidden layer, which processes the information and an output layer which presents the output to the external world. The arrangement of neurons in each layer is entirely dependent on the user, which depends on the problem to be modelled and studied. There are two types of networks—supervised and unsupervised.

Supervised learning neural networks involve the presentation of both input and corresponding output patterns to the neural network during training. The network learns all the patterns at the end of training and then the network is tested for its performance using patterns that were not used for training. The multilayer perceptron (MLP) network trained using back-propagation (BP) algorithm is a widely used network type and is commonly applied to all kinds of industrial as well as research modelling problems. Radial basis function (RBF) neural networks are a new class of robust neural network that has been used to a limited extent in modelling various research problems. Algorithms based on this type of network has been used to build neural network architecture dynamically during training and one such network type is the RAN, which builds its own architecture during training [16,17].

### 2.2.1. Multilayer perceptron (MLP)

MLP have been applied successfully to solve some difficult and diverse problems by training them in a supervised manner with a popular algorithm known as BP algorithm [15]. This algorithm allows experiential acquisition of input/output mapping knowledge within the networks. There are basically two passes through the different layers of the network: a forward pass, in which an input pattern is submitted and propagated through the
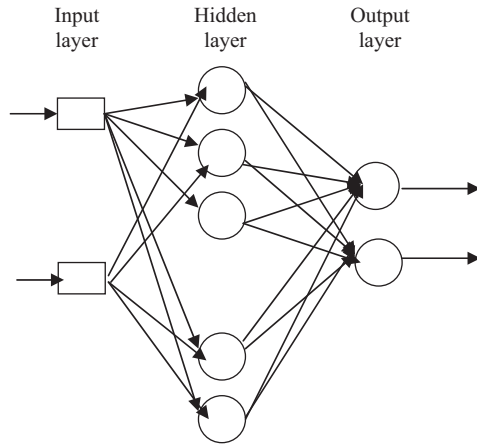
Fig. 4. MLP neural network architecture.

network layer by layer and an output is produced as the actual response of the network. During the forward pass, the synaptic weights are all fixed and in the backward pass, the synaptic weights are adjusted depending upon the error between the actual output and the desired output [16].

Fig. 4 shows the architecture of the network. There are three layers—an input, hidden and an output layer. The input layer contains two neurons corresponding to potential and load for one network and pH and load for another network. The number of neurons in the hidden layer is fixed based on the desired performance from the network. There are two output neurons corresponding to $K_{ac}$ and $K_a$. The hidden neurons and output neurons have a sigmoidal nonlinearity defined by the logistic function:

$$y_j = 1/(1 + \exp(-v_j)), \tag{2}$$

where $v_j = \Sigma w_{ji} \xi_i$ is the net internal activity of neuron $j$, $w_{ji}$ are the weights between neuron $j$ in the hidden layer and neuron $i$ in the input layer and $\xi_i$ is the input and $y_j$ is the output of the neuron. The training of the network involves propagating all the input patterns through the network layer by layer (called an 'epoch') and then in the backward pass modifying the weights based on the cumulative error computed over the entire training set. The modification of the weights is done as follows:

$$w_{ji}^{new} = w_{ji}^{old} + \Delta w_{ji}, \tag{3}$$

where $\Delta w_{ji} = \eta \alpha \Sigma \delta_j v_j$ and where $\delta_j$ is determined by partial differentiation of the total mean square error with respect to the weights. This is performed on the weights between input and hidden and hidden and output layers. $\eta$ is the learning rate, which controls the effectiveness and convergence of the algorithm and $\alpha$ is the momentum term, which is generally used to accelerate the convergence of the error BP algorithm.

### 2.2.2. Data set

There are two data sets. The first set consists of data corresponding to weight change for the microabrasion–

corrosion interaction of polymer/mild steel couple at various potentials: (a) $-0.6$ V, (b) $-0.4$ V, (c) $-0.2$ V, (d) 0 V, (e) $+0.2$ V (refer Table 3). The training data consisted of 20 patterns and the test data five patterns. The test data contains data not used for training the network. The second data set consisted of data corresponding to weight change for the microabrasion–corrosion interaction of ceramic–Ni–Cr–WC–lasercarb coating, at fixed potential $-0.2$ V and various pH (refer Table 4). The training data consisted of 16 patterns and the test data four patterns. The data have been normalized using the method

$$x_{normalized} = x/x_{max}, \tag{4}$$

where $x$ is the input pattern and $x_{max}$ is the maximum value of the pattern.

The number of neurons in the hidden layer has been varied followed by the values of $\eta$ and $\alpha$ for desirable performance in terms of minimum training time and the error reached. The optimum values for $\eta$ and $\alpha$ have been found to be 0.8. And the corresponding number of hidden neurons was found to be 15. The network converged to an error of 0.01 in 150,000 epochs.

Similarly the MLP network was trained for the second data set and the values of the parameters for optimum performance were as follows—hidden neurons $-12$, $\eta = 0.95$ and $\alpha = 0.05$. The network converged to an error of 0.03 in 125,000 epochs.

### 2.2.3. Resource allocation network

RAN is a sequential learning Gaussian Radial Basis Function network. Platt's [17] motivation for RAN stemmed from the fact that learning with a fixed-size network is a NP-complete problem (NP stands for nondeterministic polynomial) and by allocating new resources, learning could be achieved in polynomial time. The network learns by allocating new hidden neurons and adjusting the parameters of the existing neurons. If the network performs poorly on a presented pattern, then a new neuron is allocated which corrects the response to the presented pattern. Conversely, if the network performs well on a presented pattern, then the network parameters are updated using the standard least mean square (LMS) gradient descent. The structure of RAN is the same as that of RBF networks, namely an input layer, a hidden layer and an output layer. Each neuron in the hidden layer in the network has two parameters namely a center $x_j$ and width $\sigma_j$ associated with it. The activation function of the hidden neurons is radially symmetric in the input space and the output of each hidden unit depends only on the radial distance between the input vector $\xi_i$ and the center parameter $x_j$ for that hidden neuron. The response of each hidden unit is scaled by its connecting weights $w_{kj}$ to the output units and then summed to produce the overall network output. The overall network output is

$$O_k = \sum_j w_{kj} v_j, \tag{5}$$

Table 3
Weight change data for the microabrasion–corrosion interaction of polymer/mild steel couple at selected load and potentials: (a) $-0.6$ V, (b) $-0.4$ V, (c) $-0.2$ V, (d) 0 V, (e) $+0.2$ V

| Load (N) | $K_{ac}$ (g) | $K_c$ (g) | $K_a$ (g) | $K_{ao}$ (g) (weight loss at $-0.96$ V) | $\Delta K_a$ (g) | $K_c/K_a$ |
|---|---|---|---|---|---|---|
| *(a) $-0.6$ V* | | | | | | |
| 1 | 1.40E$-$04 | 1.03E$-$04 | 3.70E$-$05 | 1.3E$-$04 | $-9.30$E$-$05 | 2.78 |
| 2 | 1.90E$-$04 | 1.07E$-$04 | 8.30E$-$05 | 2.1E$-$04 | $-1.27$E$-$04 | 1.29 |
| 3 | 2.30E$-$04 | 1.03E$-$04 | 1.27E$-$04 | 2.7E$-$04 | $-1.43$E$-$04 | 0.81 |
| 4 | 4.10E$-$04 | 1.19E$-$04 | 2.91E$-$04 | 1.6E$-$04 | 1.31E$-$04 | 0.41 |
| 5 | 6.00E$-$04 | 1.36E$-$04 | 4.64E$-$04 | 2.0E$-$04 | 2.64E$-$04 | 0.29 |
| *(b) $-0.4$ V* | | | | | | |
| 1 | 2.10E$-$04 | 1.29E$-$04 | 8.10E$-$05 | 1.3E$-$04 | $-4.90$E$-$05 | 1.59 |
| 2 | 3.30E$-$04 | 1.50E$-$04 | 1.80E$-$04 | 2.1E$-$04 | $-3.00$E$-$05 | 0.83 |
| 3 | 6.40E$-$04 | 1.29E$-$04 | 5.11E$-$04 | 2.7E$-$04 | 2.41E$-$04 | 0.25 |
| 4 | 6.10E$-$04 | 1.14E$-$04 | 4.96E$-$04 | 1.6E$-$04 | 3.36E$-$04 | 0.23 |
| 5 | 7.00E$-$04 | 1.64E$-$04 | 5.36E$-$04 | 2.0E$-$04 | 3.36E$-$04 | 0.31 |
| *(c) $-0.2$ V* | | | | | | |
| 1 | 1.50E$-$04 | 1.27E$-$04 | 2.30E$-$05 | 1.3E$-$04 | $-1.07$E$-$04 | 5.52 |
| 2 | 2.00E$-$04 | 1.22E$-$04 | 7.80E$-$05 | 2.1E$-$04 | $-1.32$E$-$04 | 1.56 |
| 3 | 2.20E$-$04 | 1.27E$-$04 | 9.30E$-$05 | 2.7E$-$04 | $-1.77$E$-$04 | 1.37 |
| 4 | 2.60E$-$04 | 1.34E$-$04 | 1.26E$-$04 | 1.6E$-$04 | $-3.40$E$-$05 | 1.06 |
| 5 | 3.90E$-$04 | 1.76E$-$04 | 2.14E$-$04 | 2.0E$-$04 | 1.40E$-$05 | 0.82 |
| *(d) 0 V* | | | | | | |
| 1 | 1.30E$-$04 | 1.23E$-$04 | 7.00E$-$06 | 1.3E$-$04 | $-1.23$E$-$04 | 17.57 |
| 2 | 4.80E$-$04 | 1.25E$-$04 | 3.55E$-$04 | 2.1E$-$04 | 1.45E$-$04 | 0.35 |
| 3 | 5.50E$-$04 | 1.23E$-$04 | 4.27E$-$04 | 2.7E$-$04 | 1.57E$-$04 | 0.29 |
| 4 | 8.30E$-$04 | 1.25E$-$04 | 7.05E$-$04 | 1.6E$-$04 | 5.45E$-$04 | 0.18 |
| 5 | 1.10E$-$03 | 2.02E$-$04 | 8.98E$-$04 | 2.0E$-$04 | 6.98E$-$04 | 0.22 |
| *(e) $+0.2$ V* | | | | | | |
| 1 | 2.00E$-$04 | 9.67E$-$05 | 1.03E$-$04 | 1.3E$-$04 | $-2.67$E$-$05 | 0.94 |
| 2 | 2.30E$-$04 | 8.64E$-$05 | 1.44E$-$04 | 2.1E$-$04 | $-6.60$E$-$05 | 0.60 |
| 3 | 3.70E$-$04 | 9.68E$-$05 | 2.73E$-$04 | 2.7E$-$04 | 3.00E$-$06 | 0.35 |
| 4 | 3.70E$-$04 | 1.11E$-$04 | 2.59E$-$04 | 1.6E$-$04 | 9.90E$-$05 | 0.43 |
| 5 | 3.90E$-$04 | 1.70E$-$04 | 2.20E$-$04 | 2.0E$-$04 | 2.00E$-$05 | 0.77 |

where $j = 1$ to $c$ are the number of hidden units or neurons and $v_j = e^{-||x_j - \xi_i||^2/2\sigma^2}$, where $v_j$ is the response of the $j$th hidden unit and $w_{kj}$ is the weight connecting hidden neuron $j$ to output neuron $k$.

The learning phase involves allocation of new hidden units as well as adaptation of network parameters. The network begins with no hidden units. As input–output data are received during training, some of them are used for generating new hidden units. The decision as to whether an input–output pair $(\xi_i, \zeta_k)$ should give rise to a new hidden unit depends on the novelty in the input the data which is decided using the following two conditions: (i) $d = ||x_j - \xi_i|| > \delta$, (ii) $e = ||\zeta_k - O_k|| > e_{min}$, where $\delta$ and $e_{min}$ are thresholds to be selected appropriately. If the above two conditions are satisfied, then the data is deemed to have a novelty and a 'new hidden unit' is added. The first condition says that the input must be far away from all the centers and the second condition says that the error between the network output and target output must be significant. $e_{min}$ represents the desired approximation accuracy of the network output and the

distance $\delta$ represents the scale of resolution in the input space [18].

The algorithm begins with $\delta = \delta_{max}$, where $\delta_{max}$ is chosen as the largest scale of interest in the input space, typically the entire input space of nonzero probability. The distance $\delta$ is decayed exponentially $\delta = \max(\delta_{max}e^{(-t/\tau)}, \delta_{min})$. The value of $\delta$ is decayed until it reaches $\delta_{min}$, which is the smallest length scale of interest. This exponential decaying of the distance criterion allows fewer basis functions with large widths initially and with increasing number of observations, more basis functions with smaller widths are allocated to fine tune the approximation. The parameters associated with the new hidden units are as follows: $w_{kj}^{new} = e$, $x_j^{new} = x_j$ and $\sigma_j^{new} = \kappa ||x_j - \xi_i||$, where $\kappa$ is an overlap factor that determines the amount of overlap of the responses of the hidden units in the input space. As $\kappa$ grows larger, the responses of the neurons overlap more and more. When an input–output pair does not pass the novelty criteria, a hidden unit is not added but the network parameters $x_j$ and $w_{kj}$ are adapted to fit the observation.

Table 4
Weight change data for microabrasion–corrosion interaction of ceramic/lasercarb coating, at a fixed potential −0.2 V and selected load and pHs, (a) pH 2, (b) pH 5.5, (c) pH 9.7, (d) pH 13.5

| Load (N) | $K_{ac}$ (g) | $K_c$ (g) | $K_a$ (g) | $K_{ao}$ (g) (weight loss at −1.0 V) | $\Delta K_a$ (g) | $K_c/K_a$ |
|---|---|---|---|---|---|---|
| *(a) pH 2* | | | | | | |
| 1 | 8.87E−06 | 8.37E−06 | 5.00E−07 | 1.20E−05 | −1.15E−05 | 16.74 |
| 2 | 1.32E−05 | 7.84E−06 | 5.36E−06 | 1.30E−05 | −7.64E−06 | 1.46 |
| 3 | 1.63E−05 | 7.59E−06 | 8.71E−06 | 3.22E−05 | −2.34E−05 | 0.87 |
| 4 | 1.02E−05 | 7.04E−06 | 3.16E−06 | 3.32E−05 | −3.00E−05 | 2.23 |
| 5 | 1.09E−05 | 7.71E−06 | 3.19E−06 | 4.92E−05 | −4.60E−05 | 2.42 |
| *(b) pH 5.5* | | | | | | |
| 1 | 1.32E−05 | 7.91E−06 | 5.29E−06 | 5.36E−06 | −7.00E−08 | 1.50 |
| 2 | 1.49E−05 | 6.99E−06 | 7.91E−06 | 2.73E−05 | −1.94E−05 | 0.88 |
| 3 | 1.01E−05 | 6.34E−06 | 3.76E−06 | 3.11E−05 | −2.73E−05 | 1.69 |
| 4 | 1.48E−05 | 6.08E−06 | 8.72E−06 | 1.14E−05 | −2.68E−06 | 0.70 |
| 5 | 1.89E−05 | 6.17E−06 | 1.27E−05 | 2.47E−05 | −1.20E−05 | 0.48 |
| *(c) pH 9.7* | | | | | | |
| 1 | 3.84E−05 | 8.24E−06 | 3.02E−05 | 3.22E−05 | −2.04E−06 | 0.27 |
| 2 | 3.72E−05 | 7.95E−06 | 2.93E−05 | 3.36E−05 | −4.35E−06 | 0.27 |
| 3 | 4.04E−05 | 8.22E−06 | 3.22E−05 | 3.19E−05 | 2.80E−07 | 0.26 |
| 4 | 3.13E−05 | 8.84E−06 | 2.25E−05 | 2.88E−05 | −6.34E−06 | 0.39 |
| 5 | 4.16E−05 | 8.44E−06 | 3.32E−05 | 3.19E−05 | 1.26E−06 | 0.25 |
| *(d) pH 13.5* | | | | | | |
| 1 | 9.73E−06 | 8.09E−06 | 1.64E−06 | 2.35E−05 | −2.19E−05 | 4.93 |
| 2 | 1.03E−05 | 8.32E−06 | 1.98E−06 | 2.29E−05 | −2.09E−05 | 4.20 |
| 3 | 1.08E−05 | 7.90E−06 | 2.90E−06 | 2.36E−05 | −2.07E−05 | 2.72 |
| 4 | 1.25E−05 | 7.17E−06 | 5.33E−06 | 1.04E−05 | −5.07E−06 | 1.35 |
| 5 | 1.29E−05 | 8.02E−06 | 4.88E−06 | 2.51E−05 | −2.02 E−05 | 1.64 |
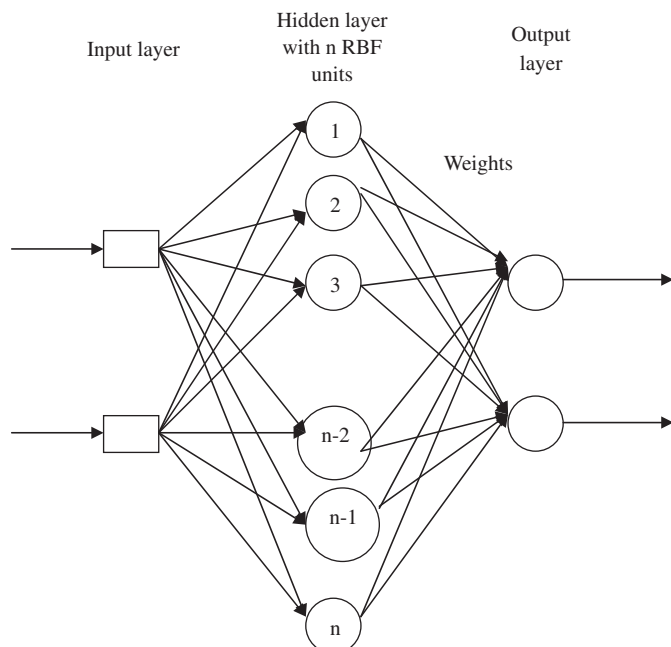


Fig. 5. RAN architecture.

The width parameter is not adapted [18]. Fig. 5 shows the RAN architecture. The data sets used above have been used to train and test RAN.

## 3. Results and discussion

### 3.1. Estimation of weight change data

The weight change data may be explained by defining the following terms, using the methodology developed by Yue and Shi [19] in wear analysis and the estimation of weight changes during microabrasion–corrosion in previous work [11]. If

$$K_{ac} = K_a + K_c, \tag{6}$$

where $K_{ac}$ is the total microabrasion–corrosion weight loss, $K_a$ is the total microabrasion weight loss, and $K_c$ is the total corrosion weight loss.

$K_a$ can be written as

$$K_a = K_{ao} + \Delta K_a, \tag{7}$$

where $K_{ao}$ is the microabrasion weight loss in the absence of corrosion, $\Delta K_a$ is the effect of corrosion on microabrasion.

$K_c$ can be explained as

$$K_c = K_{co} + \Delta K_c, \tag{8}$$

where $K_{co}$ is the corrosion weight loss in the absence of wear, $\Delta K_c$ is the effect of microabrasion on corrosion,

or the enhancement of corrosion due to the microabrasion process.

Hence, the total microabrasion–corrosion weight loss can be given as follows:

$$K_{ac} = K_{ao} + \Delta K_a + K_{co} + \Delta K_c. \tag{9}$$

The results of the various contributions to weight change are given in Tables 3 and 4. The corrosion weight loss data, $K_c$, were derived using Faraday's law, e.g.

$$K_c = Q\,(nF)^{-1}, \tag{10}$$

$$K_c = MIt\,(nF)^{-1}, \tag{11}$$

where $Q$ is the charge passed, $F$ is Faraday's constant $(96\,500\,\mathrm{C\,mol^{-1}})$, $n$ is the number of electrons involved in corrosion process, $I$ the total current, $t$ the exposure time and $M$ is the atomic mass of the material. The weight loss due to wear in the absence of corrosion, $K_{ao}$, was estimated by measuring the weight change in cathodic conditions i.e. at $-0.96\,\mathrm{V}$.

### 3.2. MLP modelling results

MLP has been used to model the microabrasion–corrosion process of steel/polymer couple and a ceramic/lasercarb coating. The basic objective of modelling is to estimate $K_{ac}$ and $K_a$ using the network and compare it with experimental results. Table 5 gives the sample results of

Table 5
Estimation of $K_{ac}$ and $K_a$ by MLP (sample results)

| Sl. No. | $K_{ac}$ (expt.) | $K_a$ (expt.) | $K_{ac}$ (network) | $K_a$ (network) | Total error |
|---|---|---|---|---|---|
| *(a) Data set 1* | | | | | |
| Training data | | | | | |
| 1 | 0.32 | 0.79 | 0.30 | 0.79 | 1.05E−05 |
| 2 | 1.00 | 1.00 | 0.97 | 0.96 | 1.42E−03 |
| 3 | 0.87 | 0.70 | 0.92 | 0.67 | 1.45E−03 |
| 4 | 0.51 | 0.69 | 0.52 | 0.69 | 4.13E−05 |
| 5 | 0.51 | 0.57 | 0.51 | 0.57 | 5.37E−06 |
| Test data | | | | | |
| 1 | 0.23 | 0.76 | 0.02 | 0.52 | 0.05 |
| 2 | 0.47 | 0.92 | 0.87 | 0.86 | 0.08 |
| 3 | 0.56 | 0.72 | 0.25 | 0.39 | 0.10 |
| 4 | 1.00 | 0.62 | 0.44 | 0.86 | 0.19 |
| 5 | 1.00 | 1.00 | 0.99 | 0.82 | 0.02 |
| *(b) Data set 2* | | | | | |
| Training data | | | | | |
| 1 | 0.81 | 0.94 | 0.82 | 0.90 | 5.73E−04 |
| 2 | 0.67 | 0.92 | 0.67 | 0.89 | 4.20E−04 |
| 3 | 0.78 | 0.77 | 0.79 | 0.76 | 9.36E−05 |
| 4 | 0.89 | 0.90 | 0.89 | 0.96 | 2.19E−03 |
| 5 | 1.00 | 0.96 | 0.94 | 0.96 | 1.64E−03 |
| Test data | | | | | |
| 1 | 0.54 | 1.00 | 0.05 | 0.89 | 0.13 |
| 2 | 0.97 | 0.93 | 0.64 | 0.93 | 0.05 |
| 3 | 0.79 | 0.88 | 0.91 | 0.96 | 0.01 |
| 4 | 0.97 | 0.86 | 0.91 | 0.97 | 6.81E−03 |

estimation for both the data sets. The average total error on training data and test data set is 5.65E−4 and 8.8E−2 for data sets 1 and 2.1E−3 and 5.0E−2 for data set 2, respectively. Hence the network has been able to estimate $K_{ac}$ and $K_a$ values effectively.

The results indicate that the estimation of values from the test data is acceptable for the purpose of evaluation. The MLP network has a good generalization capability and is able to estimate the values of $K_{ac}$ and $K_a$ within about 90% accuracy. One of the reasons for the less than satisfactory performance can be attributed to inadequate and less representative training samples [16]. It is thought that with more data representing all possible values of the parameters involved, the performance will be improved.

However, the MLP has some disadvantages i.e. it requires longer training time, has problems of local minima and has a fixed architecture, where the number of hidden neurons is fixed by trial and error (which consumes a significant amount of processing time). Hence, to overcome some of the limitations of MLP, a radial basis function (RBF) based dynamic neural network algorithm namely RAN has been used for estimation of $K_{ac}$ and $K_a$.

### 3.3. RAN modelling results

RAN has been used to model the microabrasion–corrosion process for two pairs of materials as given above. The objective is to compare the performance of this network with MLP in estimating the values of $K_{ac}$ and $K_a$. Several trials have been carried out to study the effect of various simulation parameters namely $\delta_{max}$, $\kappa$ and $\alpha$, in order to generate a network architecture which is able to estimate $K_{ac}$ and $K_a$ with minimum error. Table 6 gives the optimum simulation parameters for both data sets for desirable performance. Figs. 6 and 7 show the growth pattern for RAN as it learns sequentially from the data for both data sets 1 and 2, respectively. In Fig. 6, the addition of the hidden neurons takes place in different stages i.e. in the first four epochs four neurons are added, in the next 16 epochs, five neurons are added, from 21st to 32nd epoch, nine neurons are added and there afterwards 20 neurons are added. Whereas in Fig. 7, the network adds 16 hidden neurons in the first epoch itself and thereafter it remains constant for the remaining five epochs. It is clear that RAN

Table 6
Simulation parameters for RAN

| | Data set 1 | Data set 2 |
|---|---|---|
| $e_{min}$ | 5.00E−5 | 5.00E−5 |
| $\delta_{max}$ | 0.5 | 0.50 |
| $\delta_{min}$ | 0.01 | 0.10 |
| $\kappa$ | 0.20 | 0.50 |
| $\tau$ | 700 | 100 |
| $\alpha$ | 0.10 | 0.5 |
| No. of epochs | 62 | 6 |
| No. of RBF units (hidden neurons) | 20 | 16 |
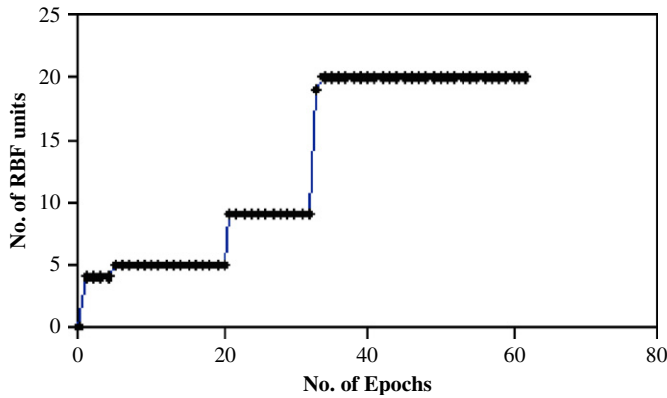
Fig. 6. Growth pattern of RAN for data set 1.



Fig. 7. Growth pattern of RAN for data set 2.

Table 7
Estimation of $K_{ac}$ and $K_a$ by RAN (sample results)

| Sl. no. | $K_{ac}$ (expt.) | $K_a$ (expt.) | $K_{ac}$ (network) | $K_a$ (network) | Total error |
|---|---|---|---|---|---|
| *(a) Data set 1* | | | | | |
| Training data | | | | | |
| 1 | 0.32 | 0.79 | 0.32 | 0.79 | 2.23E−06 |
| 2 | 1.00 | 1.00 | 1.00 | 1.00 | 6.28E−06 |
| 3 | 0.87 | 0.70 | 0.87 | 0.70 | 1.87E−06 |
| 4 | 0.51 | 0.69 | 0.51 | 0.69 | 4.38E−07 |
| 5 | 0.51 | 0.57 | 0.51 | 0.57 | 6.82E−08 |
| Test data | | | | | |
| 1 | 0.23 | 0.76 | 0.65 | 0.69 | 0.18 |
| 2 | 0.47 | 0.92 | 0.65 | 0.69 | 0.08 |
| 3 | 0.56 | 0.72 | 0.66 | 0.69 | 0.01 |
| 4 | 1.00 | 0.62 | 0.63 | 0.71 | 0.15 |
| 5 | 1.00 | 1.00 | 0.63 | 0.71 | 0.22 |
| *(b) Data set 2* | | | | | |
| Training data | | | | | |
| 1 | 0.81 | 0.94 | 0.81 | 0.94 | 1.72E−06 |
| 2 | 0.67 | 0.92 | 0.67 | 0.92 | 2.50E−06 |
| 3 | 0.78 | 0.77 | 0.78 | 0.77 | 9.17E−06 |
| 4 | 0.89 | 0.90 | 0.89 | 0.90 | 3.11E−05 |
| 5 | 1.00 | 0.96 | 0.99 | 0.96 | 3.55E−13 |
| Test data | | | | | |
| 1 | 0.54 | 1.00 | 0.01 | 0.01 | 1.26 |
| 2 | 0.97 | 0.93 | 0.63 | 0.92 | 0.12 |
| 3 | 0.79 | 0.88 | 0.60 | 0.69 | 0.07 |
| 4 | 0.97 | 0.86 | 0.43 | 0.44 | 0.47 |

adds hidden neurons almost equal to the number of patterns in the training set.

Table 7 gives the sample results of estimation for both the data sets. The average error for training data for both the data sets is almost negligible, meaning the network architecture is able to recognize all the data (100% accuracy). For test data, the error is 0.13 for data set 1 and 0.48 for data set 2. The network has been able to estimate all the values in the training data set, but the performance on the test data is unsatisfactory with errors as high as 0.48. One of the reasons for the less than satisfactory performance on the test data set can be attributed to inadequate and less representative training samples [16]. With more data representing all possible values of the parameters, the performance can be improved.

### 3.4. Comparison of results of MLP and RAN

MLP is a widely used neural network architecture used in various applications, including tribology. MLP is widely used for modelling of processes. It offers a continuous approximation of a multivariable function, that is not analytically obtainable, but that is properly described by experimental data [20]. In this study MLP has been used for estimation of $K_{ac}$ and $K_a$ and its performance has been compared with RAN, which is relatively a new neural network model used for estimation/modelling purposes.

RAN has poor generalization capability when compared to MLP because it memorizes the input data. MLP requires longer training time because it has the problem of local minima, since the learning process follows the gradient descent method. RAN has faster learning ability and produces small approximation errors when compared to MLP, but the hidden neurons added are more and is almost equal to the number of training patterns. This is the main limitation of RAN, as it generates a large network [16,17].

### 3.5. Limitation and scope

The results obviously demonstrated that ANNs can be effectively used to model tribocorrosion processes (microabrasion–corrosion). However, there are many limitations in employing such a modelling approach to a complex system like tribocorrosion, where many parameters are involved and there is huge expected influence of their interplay in various levels/measures on the entire process. Further, as the effectiveness/robustness of the ANN models mainly depend on the amount of the data, there is a need for conducting further experiments to increase the available data and also by considering different parameters. In future, ANN modelling methodology will definitely act as a tool to meet the rising need of understanding the tribocorrosion processes/mechanisms in different systems.

## 4. Conclusions

The following conclusions can be drawn from the current study:

(i) An attempt has been made to use ANNs to model the microabrasion–corrosion processes of two tribological couples, a polymer/mild steel (as a function of load and potentials and in a carbonate/bicarbonate solution of pH 9.8) and ceramic/lasercarb coating (as a function of load and pHs of different solutions).

(ii) MLP, which is used widely for prediction and classification, has been applied to model this process and estimate the values of $K_{ac}$ and $K_a$, two important parameters associated with the microabrasion–corrosion process.

(iii) The results obtained using MLP have been compared with RAN, which is a RBF based network.

(iv) The less than satisfactory performance on test data of both MLP and RAN networks indicate that there are inadequate and less representative data samples. With more data samples, the performance of the networks can be definitely improved.

(v) The results clearly demonstrate that ANNs can be effectively used to model tribocorrosion processes. There is a need for conducting further experiments and data acquisition in order to model the process in a more effective manner.

## Acknowledgement

## References

[1] Landolt D, Mischler S, Stemp M. Electrochemical methods in tribocorrosion: a critical appraisal. Electrochim Acta 2001;46:24–5.

[2] Ferreira SC, Ariza E, Rocha LA, Gomes JR, Carvalho Ph, Vaz F, et al. Tribocorrosion behaviour of $ZrO_xN_y$ thin films for decorative applications. Surf Coat Technol 2005;200:6634–9.

[3] Stack MM, Chi K. Mapping sliding wear of steels in aqueous condition. Wear 2003;255:456–65.

[4] Trezona RI, Alsopp DN, Hutchings IM. Transitions between two-body and three-body abrasive wear: influence of test conditions in the microscale abrasive wear test. Wear 1999;225–229:205–14.

[5] Rutherford KL, Hutchings IM. Theory and application of a microscale abrasive wear test. J Testing Eval JTEVA 1997;25(2):250–60.

[6] Stack MM, Mathew MT. Mapping the microabrasion resistance of WC/Co based coatings in aqueous conditions. Surf Coat Technol 2004;183(2–3):337–46.

[7] Adachi K, Hutchings IM. Wear mode mapping for the microscale abrasion test. Wear 2003;255:23–9.

[8] Stack MM, Mathew MT. Micro-abrasion transitions of metallic materials. Wear 2003;255:14–22.

[9] Buchanan FJ, Shipway PH. Microabrasion—a simple method to assess surface degradation of UHMWPE following sterilization and ageing. Biomaterials 2002;23:93–100.

[10] Gant AJ, Gee MG, May AT. Micro-abrasion of WC-Co hard metals in corrosive media. Wear 2004;256:954–62.

[11] Stack MM, Mathew MT, Jawan H. On the construction of micro-abrasion maps for a steels/polymer couple in corrosive environment. Tribol Int 2005;38(9):848–56.

[12] Tipper JL, Firkins PJ, Besong AA, Barbour PSM, Nevelos J, Stone MH, et al. Characterization of wear debris from UHMWPE on zirconia, metal-on-metal and alumina ceramic-on-ceramic hip prostheses generated in a physiological anatomical hip joint simulator. Wear 2001;250(1–12):120–8.

[13] Ingram J, Matthews JB, Pipper J, Stone M, Fisher J. Comparison of the biological activity of grade GUR 1120 and GUR 415HP UHMWPE wear debris. Bio Med Mater Eng 2002;12(2):177–88.

[14] Mindora RIPA, Laurentiu FRANGU. A survey of artificial neural networks applications in wear and manufacturing processes. The Annals of University "DUNAREA DE JOS" of GALATI, FASCICLE VIII, ISSN 1221-4590, TRIBOLOGY 2004; 35–42.

[15] Zhang Z, Friedrich K. Artificial neural networks applied to polymer composites: a review. Compos Sci Technol 2003;63:2029–44.

[16] Srinivasa PP. Acoustic emission based tool wear monitoring using some improved neural network methodologies. Ph.D. thesis, S.J. College of Engineering, University of Mysore, Mysore; 2004.

[17] Platt J. A resource allocating network for function interpolation. Neural Comput 1991;3:213–25.

[18] Yingwei L, Sundarajan N, Saratchandran P. Performance evaluation of a sequential minimal RBF neural network learning algorithm. IEEE Trans Neural Networks 1998;9(2):308–18.

[19] Yue Z, Zhou P, Shi J. In: Luedema KC, editor. Proceedings of Conference on Wear of Materials. New York: ASME; 1987. p. 763–8.

[20] Laurentiu FRANGU, Minodora RIPA. Artificial neural networks applications in tribology—a survey. NIMIA-SC 2001-2001 NATO Advanced Study Institute on Neural Networks for Instrumentation, Measurement and Related Industrial Applications: Study Cases, Crema, Italy, 9–20 October 2001.