Int J Adv Manuf Technol (2008) 39:1101–1110
DOI 10.1007/s00170-007-1304-5

ORIGINAL ARTICLE

Artificial neural network based on genetic learning for machining of polyetheretherketone composite materials

C. A. Conceição António · J. Paulo Davim · Vítor Lapa

Received: 26 June 2006 / Accepted: 2 November 2007 / Published online: 1 December 2007

Abstract In this paper an artificial neural network (ANN) aiming for the efficient modelling of a set of machining conditions for orthogonal cutting of polyetheretherketone (PEEK) composite materials is presented. The supervised learning of the ANN is based on a genetic algorithm (GA) supported by an elitist strategy. Input, hidden and output layers model the topology of the ANN. The weights of the synapses and the biases for hidden and output nodes are used as design variables in the ANN learning process. Considering a set of experimental data, the mean relative error between experimental and numerical results is used to monitor the learning process obtaining the completeness of the machining process modelling. Also a regularization term associated to biases in hidden and output neurons are included in the GA fitness function for learning. Using a different set of experimental results, the optimal ANN obtained after learning is tested. The optimal number of nodes on the hidden layer is searched and the positive influence of the regularization term is demonstrated. This approach of ANN learning based on GA presents low mean relative errors in learning and testing phases.

Keywords Polymer-matrix composites · ANN simulation · Genetic algorithm · Process monitoring · Machining

C. A. C. António (✉)
Faculty of Engineering, University of Porto,
Rua Dr. Roberto Frias, s/n,
4200-465 Porto, Portugal
e-mail: cantonio@fe.up.pt

J. P. Davim · V. Lapa
Mechanical Engineering Department, University of Aveiro,
Campus Universitário de Santiago,
3810-193 Aveiro, Portugal

J. P. Davim
e-mail: pdavim@mec.ua.pt

1 Introduction

Since machining processes are non-linear and time-dependent, it is difficult for classical identification methods to provide accurate models. To address this difficulty, non-classical methods such as artificial neural networks (ANNs) are used. The ANNs are robust models having properties of universal approximation, parallel distributed processing, learning, and adaptive behaviour and can be applied to multivariate systems. ANNs give a kind of implicit relationship between inputs and outputs by learning from a data set representing the behaviour of a system.

In the last years, modelling using artificial neural networks (ANNs) have been extensively used and investigated in machining (Liu and Altintas [1]; Lin and Ting [2]; Das et al. [3]; Choudhury et al. [4]; Briceno et al. [5]; Oktem et al. [6]; Tsai et al. [7]; and Suresh et al. [8], among others), and in other materials processing technologies (Zhecheva et al. [9]; Altinkok and Koker [10]). In particular, the ANNs have been used in machining processes such as milling, turning, and drilling. Liu and Altintas [1] developed a feed-forward neural network algorithm (MLFF N-Network) using cutting speed, feedrate, and measured cutting forces as input parameters to get on-line monitoring of flank wear in turning. Lin and Ting [2] used a back propagation neural network (BPNN) with sample and batch mode, and observed a faster convergence to minimal error in the case of the sample mode. For a neural network with two hidden layers with the same number of nodes, they also noticed that convergence is achieved faster than with one hidden layer, and they reported that at higher learning rate, error is reduced. In another work, Das et al. [3] used a back propagation algorithm for measuring the flank wear of carbide tool in turning operation. Choudhury et al. [4] developed a three-layer, feed-forward back propagation neural network for predicting flank wear in turning operations, using a geometrical relationship when correlating the flank wear on cutting tool with changes in workpiece dimensions. Briceno et al. [5] showed the feasibility of radial basis network (RBN) in the modelling process of metal cutting operations. ANN models were proposed by Oktem et al. [6], Tsai et al. [7], and Suresh et al. [8] to predict surface roughness during the end milling process. In particular, Oktem et al. [6] and Suresh et al. [8] used ANN together with genetic algorithm (GA) to minimize the surface roughness.

A literature survey of ANN applications shows that only a small number of works is devoted to orthogonal cutting and all of them are not applied to polymeric composite materials. Modelling of orthogonal cutting of composite materials is implemented in this work. By definition, the cutting is orthogonal when the tool has a position angle of 90° and an inclination angle of 0°, enabling this way a bi-dimensional representation of the plane deformation of the chip as referred by Childs et al. [11]. To explain this phenomenon, the physical-mathematical model of approximation of Merchant has been used to obtain the vector analysis of cutting forces and stresses of the machining process. However, for some applications such as in composite materials machining processes, the Merchant model is not satisfactory. Furthermore some machining parameters assume discrete values, making the use of polynomial approximations inappropriate. The proposed alternative is to use an approximation model based on ANN.

The objective of this work is to develop an ANN model based on evolutionary learning, applied to polymeric composite materials machining. Section 2 presents the model overview, the topology of ANN, the input and output parameters, the data pre-processing and the dynamics of ANN. Section 3 describes the ANN learning process based on genetic search. The learning variables, the code format, the fitness function and the genetic algorithm are the main aspects of Section 3. The results of learning procedure and the discussion of the best configuration for the ANN are presented in Section 4. The influence of the regularization term associated with biases in hidden and output nodes is studied. The testing of the optimal ANN topology is presented and discussed also in this section. Finally, the achievements obtained with the developed approach are presented in Section 5.

2 Artificial neural networks model

2.1 Model overview

In the present work, a set of experimental results in turning of polyetheretherketone (PEEK) composite materials has been obtained considering as process parameters the cutting speed, feedrate, type of insert of the tool, and type of workpiece material. Then, a set of these experimental results is used in the learning algorithm of ANNs to establish the relationship between the referred input parameters and the output machining parameters such as cutting force, feed force, and chip thickness after cutting. The learning procedure is performed as an optimization algorithm based on a genetic search. The optimal ANN is tested after the learning process using another set of experimental data. Figure 1 shows the flowchart of learning and testing procedures of the proposed ANN model.

2.2 Neural network topology

The artificial neural network (ANN) is a computational structure inspired by biological neural human systems. ANN is formed of simple and highly interconnected processors called neurons. These neurons are connected to each other by weighted links denoted by synapses. These connections establish the relationship between input data O_i^{inp} and output data O_j^{out}.

The biases in the neurons of the hidden and output layers, $b_k^{(1)}$ and $b_k^{(2)}$ respectively, are controlled during data processing. Input, hidden, and output layers form the topology of the ANN used in this work. Figure 2 shows the topology of the ANN together with the input and output parameters.

Two sigmoid activation functions are used in hidden and output layers. The relative error determined by comparing experimental and numerical results is used to monitor the learning process as an optimization process. The objective is to obtain the completeness of modelling of the machining process. A regularization function related with biases in hidden and output neurons is introduced to improve the learning process.

2.3 Input and output parameters

In order to sufficiently train the network arrays of input and output, a considerable number of data pairs is considered. Two materials are used in the experimental procedure: (1) unreinforced polyetheretherketone (PEEK) and (2) reinforced polyetheretherketone with 30% of glass fibres (PEEK GF 30) (supplied by ERTA®). Table 1 shows the mechanical and thermal properties of these materials.

The orthogonal cutting tests were carried out in extruded workpieces with a diameter of 50 mm and a length of 100 mm, using a polycrystalline (PCD) insert tool and cemented carbide (K20) tool. A type SDJCL 2020 K16 tool holder was used. The tool geometry was as follows: rake angle 0°, clearance angle 7°, cutting edge angle 91° and cutting edge inclination angle 0°.

A CNC lathe “Kingsbury MHP 50” of 18 kW spindle power and a maximum spindle speed of 4,500 rpm were used to perform the experiments. Figure 3 shows the experimental apparatus used in the process.

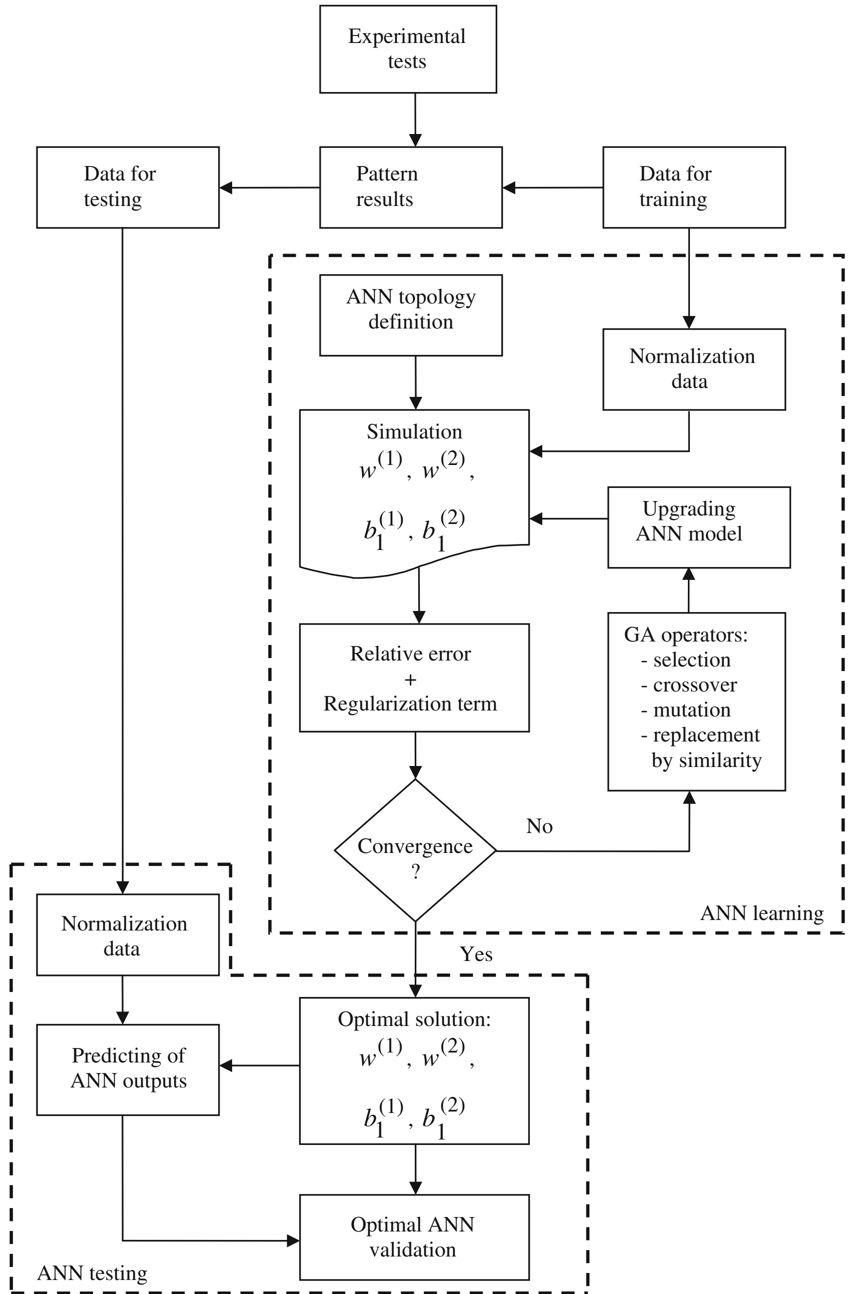

Fig. 1 ANN learning and testing flowchart

The plan of tests was developed without refrigeration and contemplates combinations between four values of cutting velocity and five values of feedrate. A constant depth of cut at 2.5 mm was considered. A Kistler® 9121 piezoelectric dynamometer with a charge amplifier (model 5019) was used to acquire the cutting forces and feed forces. Data acquisition was made through the charge amplifier and a computer using the appropriate software (Dynoware by Kistler®). The chip thickness measurement after cutting was implemented through a Mitutoyo® digital micrometer with a range of 0–25 mm and a resolution of 0.001 mm. The process parameters and respective ranges were selected according to acquired industrial experience for polymeric composite materials. Then, in the experimental tests, the following values were considered for cutting parameters: cutting speed: {80, 160, 320, 500} [m/min], feedrate: {0.05, 0.10, 0.15, 0.20, 0.30} [mm/rev].

The above values of the process parameters were combined with: Two workpiece materials: (1) unreinforced PEEK and (2) glass reinforced PEEK GF 30: Two insert tools: (1) K20 and (2) PCD to obtain the experimental results for cutting force, feed force, and chip thickness after cutting. A total number of 67 patterns have been obtained from this experimental procedure as presented in Table 2. Close to 80% of these pattern results are used in ANN learning procedure and the remaining ones used in testing the achieved optimal ANN.

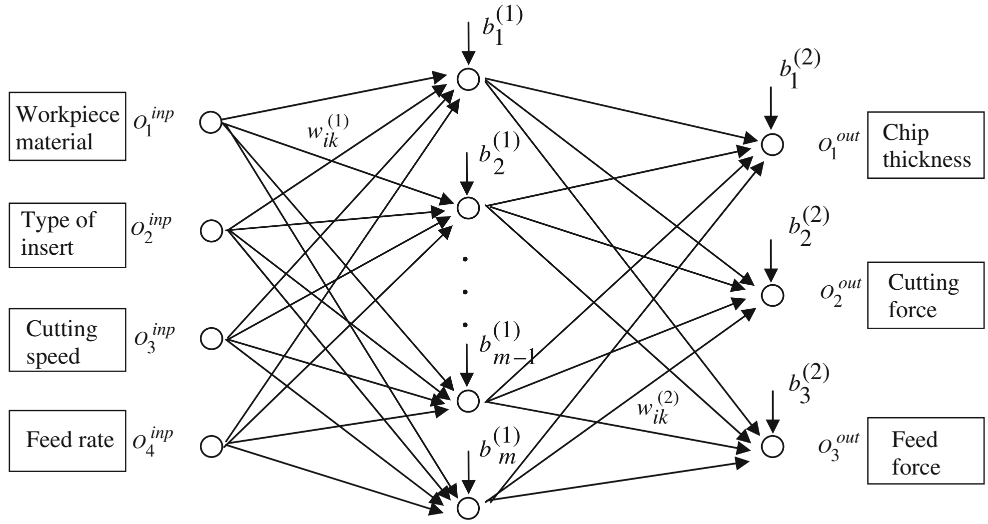

Fig. 2 Topology of artificial neural network

2.4 Data pre-processing

Each pattern consisting of an input vector and an output vector needs to be normalized to avoid error propagation during the learning process of the ANN. This is achieved using the following data normalization:

$$\overline{O}_k = (O_k - O_{\min})\frac{O_{N\max} - O_{N\min}}{O_{\max} - O_{\min}} + O_{N\min} \tag{1}$$

where O_k is the real value of the variable before normalization, O_{min} and O_{max} are the minimum and the maximum values of O_k in the data set to be normalized. Then they are normalized to values $O_{N\ min}$ and $O_{N\ max}$ such that $O_{N\ min} \leq \overline{O}_k \leq O_{N\ max}$. Depending on the input or the output data different predefined values of $O_{N\ min}$ and $O_{N\ max}$ can be used.

2.5 Dynamics of neural network

The output of the k-th neuron in the hidden layer O_k^{hid} is define as,

$$O_k^{hid} = \frac{1}{1 + \exp\left(\frac{-I_k^{hid}}{T^{(1)}}\right)} \tag{2}$$

with

$$I_k^{hid} = \sum_{j=1}^{N^{inp}} w_{jk}^{(1)} O_j^{inp} + b_k^{(1)} \tag{3}$$

where N^{inp} is the number of elements in the input, O_j^{inp} is the input data, $w_{jk}^{(1)}$ is the connection weight of the synapse between the j-th neuron in the input layer and the k-th neuron in the hidden layer, $b_k^{(1)}$ is the bias in the k-th neuron of the hidden layer and $T^{(1)}$ is a scaling parameter.

Table 1 Mechanical and thermal properties of PEEK and PEEK GF30

Mechanical and thermal properties	PEEK	PEEK GF30	Unit
Tensile modulus (E)	4,200	8,100	N/mm^2
Rockwell hardness ISO 2039-2	M105	M108	-
Elongation	20	3	%
Tensile strength	110	160	N/mm^2
Glass transition temperature (Tg)	143	143	°C
Melting temperature	340	340	°C
Density	1.31	1.5	Kg/m^3
Coefficient of thermal expansion (<150°C)	5E-05	2.5E-05	m/m.K
Coefficient of thermal expansion (>150°C)	11E-05	5.50E-05	m/m.K

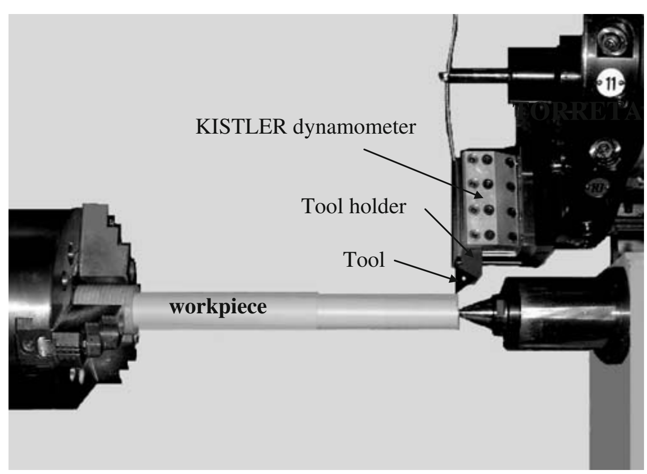

Fig. 3 Experimental apparatus

Table 2 Patterns of input/output used in learning and testing processes of ANN

Experiment number	Cutting speed [m/min]	Feedrate [mm/rev]	Insert tool	Workpiece material	Chip thickness [mm]	Cutting force [N]	Feed force [N]
1	80	0.05	K20	PEEK GF 30	0.094	33.239	55.144
2	80	0.05	PCD	PEEK GF 30	0.096	31.609	50.847
3	80	0.10	PCD	PEEK GF 30	0.184	33.280	84.480
4	80	0.15	K20	PEEK GF 30	0.258	43.331	130.065
5	80	0.15	PCD	PEEK GF 30	0.263	38.231	118.172
6	80	0.20	K20	PEEK GF 30	0.328	45.325	162.214
7	80	0.30	K20	PEEK GF 30	0.450	47.287	224.452
8	80	0.30	PCD	PEEK GF 30	0.459	43.385	208.232
9	160	0.05	K20	PEEK GF 30	0.093	31.376	53.211
10	160	0.05	PCD	PEEK GF 30	0.094	30.739	50.586
11	160	0.10	K20	PEEK GF 30	0.177	35.145	89.322
12	160	0.15	K20	PEEK GF 30	0.254	40.963	128.136
13	160	0.15	PCD	PEEK GF 30	0.257	37.389	119.466
14	160	0.20	K20	PEEK GF 30	0.322	46.132	162.378
15	160	0.20	PCD	PEEK GF 30	0.324	41.256	149.692
16	160	0.30	PCD	PEEK GF 30	0.447	42.697	206.814
17	320	0.05	K20	PEEK GF 30	0.092	30.263	52.693
18	320	0.05	PCD	PEEK GF 30	0.093	28.785	50.051
19	320	0.10	PCD	PEEK GF 30	0.177	32.142	83.217
20	320	0.15	K20	PEEK GF 30	0.252	36.870	124.021
21	320	0.20	K20	PEEK GF 30	0.322	42.117	160.526
22	320	0.30	K20	PEEK GF 30	0.438	42.961	225.594
23	80	0.05	K20	PEEK	0.071	37.504	65.157
24	80	0.05	PCD	PEEK	0.074	36.978	63.058
25	80	0.10	K20	PEEK	0.140	41.016	110.422
26	80	0.10	PCD	PEEK	0.144	40.528	106.027
27	80	0.15	K20	PEEK	0.205	43.535	154.032
28	80	0.20	K20	PEEK	0.268	45.954	197.576
29	80	0.20	PCD	PEEK	0.275	45.316	190.397
30	80	0.30	PCD	PEEK	0.396	48.284	274.746
31	160	0.05	K20	PEEK	0.070	37.409	64.205
32	160	0.10	K20	PEEK	0.138	39.221	106.212
33	160	0.10	PCD	PEEK	0.143	37.334	103.049
34	160	0.15	PCD	PEEK	0.209	39.678	143.652
35	160	0.20	K20	PEEK	0.266	42.157	191.534
36	160	0.20	PCD	PEEK	0.272	41.019	187.385
37	160	0.30	K20	PEEK	0.390	44.930	274.340
38	160	0.30	PCD	PEEK	0.393	43.698	270.094
39	320	0.05	K20	PEEK	0.069	35.957	61.830
40	320	0.10	K20	PEEK	0.136	36.006	102.898
41	320	0.10	PCD	PEEK	0.141	35.626	100.680
42	320	0.15	K20	PEEK	0.200	37.823	145.994
43	320	0.15	PCD	PEEK	0.207	36.539	141.589
44	320	0.20	K20	PEEK	0.262	38.581	187.006
45	320	0.20	PCD	PEEK	0.270	37.304	180.577
46	320	0.30	PCD	PEEK	0.390	38.919	260.073
47	500	0.05	K20	PEEK	0.068	29.674	58.148
48	500	0.05	PCD	PEEK	0.072	27.789	55.501
49	500	0.10	PCD	PEEK	0.140	29.331	97.498
50	500	0.15	K20	PEEK	0.198	32.035	142.446
51	500	0.15	PCD	PEEK	0.206	30.656	136.500
52	500	0.20	K20	PEEK	0.260	33.442	182.998
53	500	0.30	K20	PEEK	0.381	36.583	264.289
54	500	0.30	PCD	PEEK	0.387	34.189	257.752
55	80	0.10	K20	PEEK GF 30	0.180	37.285	92.714

Table 2 (continued)

Experiment number	Cutting speed [m/min]	Feedrate [mm/rev]	Insert tool	Workpiece material	Chip thickness [mm]	Cutting force [N]	Feed force [N]
56	80	0.20	PCD	PEEK GF 30	0.335	41.585	150.471
57	160	0.10	PCD	PEEK GF 30	0.179	32.191	84.045
58	160	0.30	K20	PEEK GF 30	0.444	49.061	227.572
59	320	0.10	K20	PEEK GF 30	0.176	33.706	88.746
60	80	0.15	PCD	PEEK	0.211	43.150	148.378
61	80	0.30	K20	PEEK	0.390	48.879	279.209
62	160	0.05	PCD	PEEK	0.074	36.481	61.697
63	160	0.15	K20	PEEK	0.203	40.649	147.965
64	320	0.05	PCD	PEEK	0.073	33.818	59.834
65	320	0.30	K20	PEEK	0.384	40.172	267.522
66	500	0.10	K20	PEEK	0.140	29.331	97.498
67	500	0.20	PCD	PEEK	0.267	31.898	178.558

The value of the output neuron is calculated as,

$$O_k^{out} = \frac{1}{1 + \exp\left(-\frac{I_k^{out}}{T^{(2)}}\right)} \tag{4}$$

with

$$I_k^{out} = \sum_{i=1}^{N^{hid}} w_{ik}^{(2)} O_i^{hid} + b_k^{(2)} \tag{5}$$

where N^{hid} is the number of neurons in the hidden layer, $w_{ik}^{(2)}$ is the connection weight of the synapse between the i-th neuron of the hidden layer and the k-th neuron in the output layer, $b_k^{(2)}$ is the bias in the k-th neuron of the output layer and $T^{(2)}$ is a scaling parameter for this layer.

During training, the computed output is compared with the measured output and the mean relative error is calculated as

$$E\left(w^{(1)}, w^{(2)}, b^{(1)}, b^{(2)}\right) = \frac{1}{N^{\exp}} \sum_{m=1}^{N^{\exp}} \left(\frac{1}{N^{out}} \sum_{i=1}^{N^{out}} \left| \frac{\overline{O}_i^{\exp} - \overline{O}_i^{out}}{\overline{O}_i^{\exp}} \right| \right)_m \tag{6}$$

where N^{out} is the number of neurons of the output layer, N^{exp} is the number of experimental patterns and $\overline{O}_i^{out}$ and $\overline{O}_i^{\exp}$ are the normalized predicted and measured values, respectively.

The error obtained from (6) is back propagated in to the artificial network. This means that from output to input the weights of the synapses and the biases can be modified until the error falls within a prescribed value.

3 Learning based on genetic search

Genetic algorithms (GAs) have been used with increasing interest in a large variety of applications. They search the solution space simulating the evolution of survival of the fittest according to Darwin's theory. The GAs use a structured exchange of data to explore all regions of the domain and lead some operators to exploit potential areas. Besides genetic operator's basic concepts such as chromosome representation, generation of initial population, stopping criteria and fitness function are important in GAs.

The use of GAs in ANN learning is a methodology applied by some authors such as Mok et al. [12], Nakhjavani and Ghoreishi [13] in materials processing simulation. In the present work, the objective is to explore some advantages of GAs in handling discrete input parameters associated with the machining process such as type of workpiece material and type of insert of the tool. This way the learning process becomes an optimization process.

By means of training, the neural network models the underlying process of a certain mapping. In order to model the mapping, the network weights and biases of the ANN topology have to be found. There are two categories of training algorithms: supervised and unsupervised. In this work supervised learning is used and a set of data samples called a training set for which the corresponding network outputs are known is provided. A schematic representation of the ANN learning algorithm is given in Fig. 1.

3.1 Learning variables and encoding

The supervised learning process of the ANN based on a genetic algorithm uses the weights of synapses and the bias of neural nodes of hidden and output layers as design variables. Thus, the learning variables are the weights of synapses linking the input layer nodes to hidden layer nodes $w_{jk}^{(1)}$, the weights of synapses linking hidden layer nodes to output layer nodes $w_{ik}^{(2)}$ and the bias in nodes of hidden and output layers, $b_k^{(1)}$ and $b_k^{(2)}$, respectively.

A binary code is used to encode the domain values for the learning variables. The number of digits of each variable can be different. The bounds of interval domain of learning variables influence the sensitivity of sigmoid activation functions and must be controlled. Their influence is similar to the one for scaling parameters $T^{(1)}$ and $T^{(2)}$ at hidden and output layers, respectively.

3.2 Fitness function

The proposed optimisation problem formulation is based on the minimisation of the mean relative error defined in Eq. (6). Thus in the evolutionary search it is intended to maximise a global fitness function *FIT* and the optimisation problem is defined as follows

$$Maximize\ FIT = K - \left[E\left(w^{(1)}, w^{(2)}, b^{(1)}, b^{(2)}\right) + B\left(b^{(1)}, b^{(2)}\right)\right] \tag{7}$$

where $B\left(b^{(1)}, b^{(2)}\right)$ is a regularization function associated to mean quadratic biases of hidden and output layers and defined as

$$B\left(b^{(1)}, b^{(2)}\right) = \frac{1}{N^{\exp}}\left(B^{(1)} + B(2)\right)^{1/2} \tag{8}$$

with

$$B^{(1)} = \sum_{m=1}^{N^{\exp}}\left[\frac{1}{N^{hid}}\sum_{k=1}^{N^{hid}}\left(b_k^{(1)}\right)^2\right]_m \tag{9}$$

and

$$B^{(2)} = \sum_{m=1}^{N^{\exp}}\left[\frac{1}{N^{out}}\sum_{k=1}^{N^{out}}\left(b_k^{(2)}\right)^2\right]_m \tag{10}$$

being $b_k^{(1)}$ and $b_k^{(2)}$ the associated biases in the nodes of hidden and output layers, respectively. Regularization functions are used to accelerate the convergence of the evolutionary process.

3.3 Genetic algorithm

Four main operators applied to the population support the genetic algorithm (GA): *selection*, *crossover*, *implicit mutation* and *replacement* of similar individuals as proposed by António et al. [14]. These operators are performed by an elitist strategy that always preserves a core of best individuals of the population that is transferred into the next generations. The offspring group formed by the crossover operator will make part of the population of the next generation. To avoid the rising of local minima, a chromosome set in which genes are generated randomly is introduced into the population. The genetic algorithm performs as follows:

Step 1: *Initialization*. The initial population is randomly generated.

Step 2: *Selection*. This operator chooses the population part that will be transferred into the next generation after ranking based fitness of the actual population. An elitist strategy where only the best chromosomes from the actual population will pass into the next population is adopted. The operator selects the progenitors: one from the best-fitted group (elite) and another from the least fitted one. This selection is done randomly with an equal probability for each individual. Transfer of the whole population to an intermediate step where they will join the offspring determined by the *Crossover* operator.

Step 3: *Crossover*. The offspring genetic material is obtained using a modification of the "*Parametrized Uniform Crossover*" technique proposed by Spears and DeJong [15] and modified by António and Davim [16]. This is a multipoint combination technique applied to the binary string of the selected chromosomes. The offspring gene is selected on a biased way given a defined probability for choosing the gene from the elite chromosome.

Step 4: *Replacement by similarity*. A new ranking of the enlarged population according to individual fitness is implemented. Then, it follows *Elimination* of solutions with similar genetic properties and subsequent *Substitution* by new randomly generated individuals. *Substitution* and new ranking updating is followed by *Elimination* corresponding to the deletion of the worst solutions. The exclusion of individuals with low fitness and also the natural death of old individuals are simulated by this operator. Now, the dimension of the population is smaller than the original one. The original size population will be recovered after including a group of new solutions obtained from the *Mutation* operator.

Step 5: *Implicit Mutation*. To avoid the rising of local minima, a chromosome group, where genes are generated in a random way, is introduced into the population. This operation is called implicit mutation. The number of individuals in *Implicit mutation* is an important GA parameter to keep the diversity of the population in each generation. Experience indicates that the advised number is located between 20% and 35% of the population size.

Step 6: *Stopping criterion*. The stopping criterion used in the evolutionary process is based on the relative variation of the mean fitness of a reference group during a fixed number of generations and the feasibility of the corresponding solutions. The size for the reference group is predefined as referred by António et al. [14].

The flowchart of the ANN learning and testing procedures coupled with the genetic algorithm is presented in

Fig. 1. The operating conditions of the GA have been given above and follow the model detailed in António et al. [14].

4 Training and testing the artificial neural network

This section presents the results of the learning procedure of artificial neural network (ANN) and of the additional testing phase. The experimental plan of tests was implemented and the obtained 67 patterns of results are presented in Table 2. Eighty percent of these results are used in the learning procedure and the remaining results are used to test the obtained optimal ANN topology.

To perform the data normalization it was used $O_{N\ min}$=0.1 and $O_{N\ max}$=0.9 defined in Eq. (1) for the input and output results, respectively. A code format of five digits was used for binary encoding of weights in the synapses and biases in hidden and output neurons. In the ANN learning procedure based on genetic search and also in the testing process it was considered a population of 15 individuals. Five individuals form the elite group of the population that will be transferred to next generation. In implicit mutation, five individuals are inserted into the population to keep the diversity at an acceptable level.

A study of best topology for ANN was carried out considering the variation of the number of nodes in the hidden layer. Using the mean relative error in learning and testing phases the results for different number of hidden nodes are compared and shown in Fig. 4. The best result was obtained

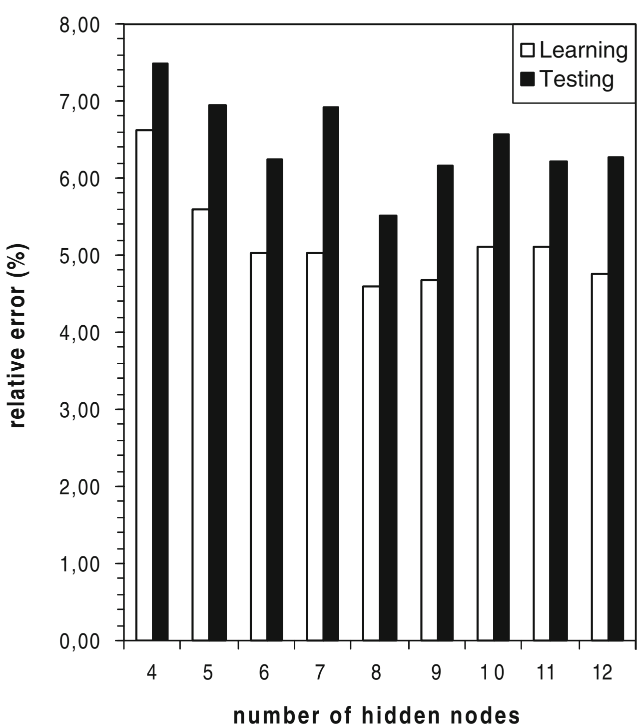

Fig. 4 Mean relative errors in learning and testing processes as function of the number of nodes of the hidden layer

considering a number of eight neurons for the hidden layer that will be used in the next presentation and discussion.

4.1 ANN learning process

The first 54 patterns of values given in Table 2 are used in the ANN training. The learning process implemented for this ANN topology evolves along 30,000 generations. The evolution processes for both components of fitness function in Eq. (7), the absolute value of the mean relative error, and the regularization term, are shown in Fig. 5.

The best result obtained for the mean relative error was 4.6%. The regularization term also reaches a minimum value and its influence can be analyzed considering the fitness function with and without regularization term. Taking only the relative error in Eq. (7) it follows

$$FIT = K - E\left(w^{(1)}, w^{(2)}, b^{(1)}, b^{(2)}\right) \tag{11}$$

If the genetic search is performed using Eq. (11), different values of mean relative error during the learning procedure are observed. The differences in the mean relative error are calculated as

$$\Delta E(\%) = (E_B - E_0) \times 100 \tag{12}$$

where E_B and E_0 are the mean relative errors obtained considering the fitness function with and without regularization term, respectively.

Figure 6 shows the influence of the regularization term through the behaviour of the difference $\Delta E(\%)$ along the learning procedure based on genetic search. At the beginning, the relative error is higher with regularization term but after 10,000 generations the situation is reverted.

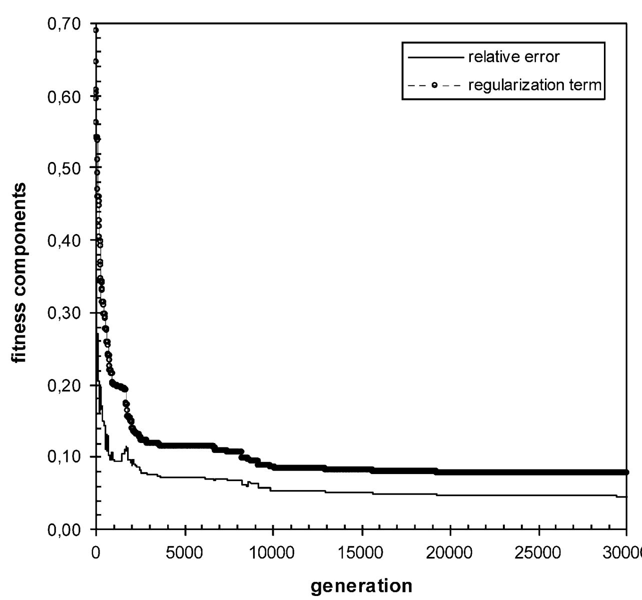

Fig. 5 Mean relative error and regularization term behaviours during ANN learning process based on genetic search

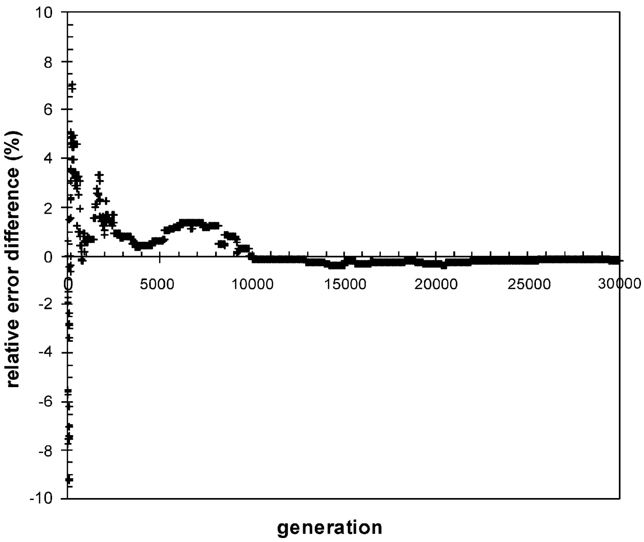

Fig. 6 Mean relative error differences *ΔE(%)* associated with regularization term in fitness function

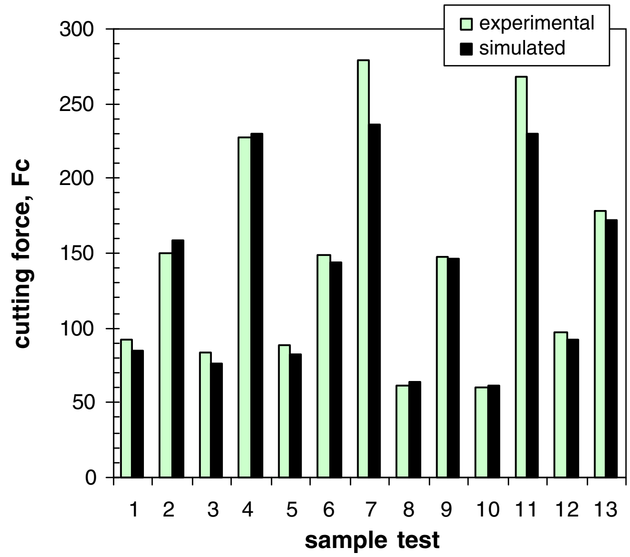

Fig. 8 Testing process of ANN with optimal topology: comparisons for cutting force, Fc (N)

Applying the learning procedure with the complete definition of fitness function given in Eq. (7) the ANN optimal topology was obtained.

4.2 ANN testing process

The last 13 patterns of results of Table 2 were not included in the ANN learning process and have been used to test and validate the ANN with the optimal topology obtained after learning procedure and with the complete definition of fitness function in Eq. (7). These input testing values were introduced in the optimal ANN and the simulated output values are compared with the experimental ones. Figures 7, 8, and 9 show the experimental and the ANN simulated outputs for the testing group considering the interaction of all parameters.

The mean relative error in this testing process is 5.5%, which is considered a good agreement between the experimental results and the simulated outputs given by the optimal ANN topology. This percentage of mean relative error corresponds to the combination of all results presented in Figs. 7, 8, and 9 and is higher than the obtained percentage at the end of the learning procedure (4.6%) considering only the data set for learning.

The benefits of the regularization term in Eq. (7) are also observed in the testing procedure when both ANNs obtained after learning using fitness functions with and

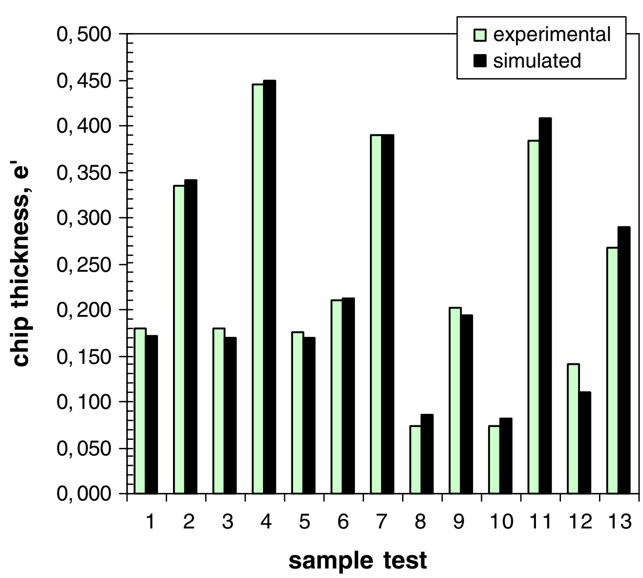

Fig. 7 Testing process of ANN with optimal topology: comparisons for chip thickness, e' (mm)

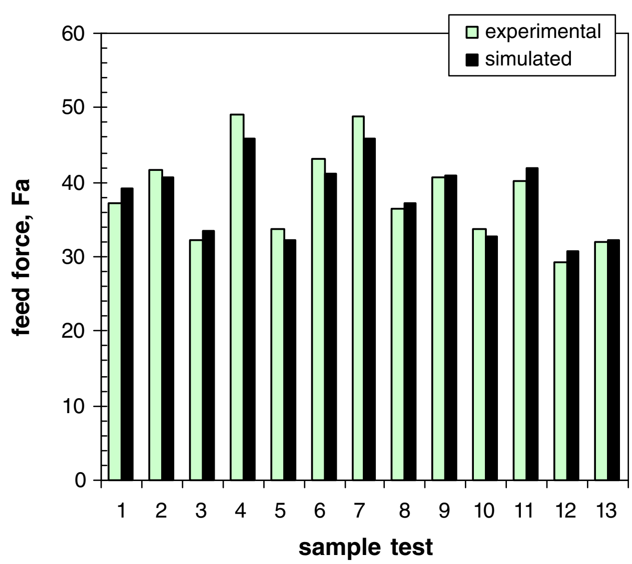

Fig. 9 Testing process of ANN with optimal topology: comparisons for feed force, Fa (N)

without regularization term are compared. The mean relative error obtained in testing procedure using the ANN based on fitness function without regularization term as defined in Eq. (11) is 6%, which is worse than the one with regularization term (5.5%).

5 Conclusions

In this paper, an artificial neural network (ANN) aiming for the efficient modelling of a set of machining conditions in orthogonal cutting of PEEK composite materials is presented. The proposed ANN is based on input, hidden, and output layers. The input parameters are cutting speed, feedrate, type of insert of the tool, and type of workpiece material. The output parameters are cutting force, feed force, and the chip thickness after cutting. Two sigmoid functions in hidden and output layers are used. The supervised learning of the ANN is based on a genetic algorithm with an elitist strategy. The mean relative error between experimental and numerical results was used to monitor the learning process.

To illustrate the computational methodology, a set of experimental results is considered and the numerical results obtained from supervised learning are presented. Then, using a different set of experimental results, the obtained solution for ANN is tested aiming to demonstrate the performance of the learning process. An additional study on the influence of the number of nodes in hidden layer for mean relative error was carried out. This study advises an ANN with eight nodes for the hidden layer.

The results obtained with the optimization process for supervised learning show that genetic search is a good option. The additional testing procedure proves that there is a good agreement between experimental results and simulated outputs obtained with the optimal ANN topology. The use of a regularization term associated with the minimization of biases in hidden and output layers induces benefits during the learning procedure based on genetic algorithm. The improvement associated with the regularization term in the optimal ANN topology was also demonstrated with the testing procedure.

References

1. Liu Q, Altintas Y (1999) On-line monitoring of flank wear in turning with multi-layered feed-forward neural network. Int J Mach Tool Manuf 39:1945–1959
2. Lin SC, Ting CJ (1996) Drill wear monitoring using neural network. Int J Mach Tool Manuf 36:465–475
3. Das S, Roy R, Chottopadhyay AB (1996) Evaluation of wear of turning carbide inserts using neuronal network. Int J Mach Tool Manuf 36:789–797
4. Choudhury SK, Jain VK, Rama Rao CVV (1999) On-line monitoring of tool wear in turning using neural network. Int J Mach Tool Manuf 39:489–504
5. Briceno J, El-Mounayri H, Mukhopadhyay S (2002) Selecting an artificial neuronal network for efficient modeling and accurate simulation of the milling process. Int J Mach Tool Manuf 42:663–674
6. Oktem H, Erzurumlu T, Erzincanli F (2006) Prediction of minimum surface roughness in end milling mold parts using neural network and genetic algorithm. Mater Design 27:735–744
7. Tsai YH, Chen JC, Lou SJ (1999) An in-process surface recognition system based on neural networks in end milling cutting operations. Int J Mach Tool Manuf 39:583–605
8. Suresh PVS, Rao PV, Deshmukh SG (2002) A genetic algorithmic approach for optimization of surface roughness prediction model. Int J Mach Tool Manuf 42:675–680
9. Zhecheva A, Malinov S, Sha W (2005) Simulation of microhardness profiles of titanium alloys after surface nitriding using artificial neural network. Surface Coat Technol 200:2332–2342
10. Altinkok N, Koker R (2006) Modelling of the prediction of tensile and density properties in particle reinforced metal matrix composites by using neural networks. Mater Design 27:625–631
11. Childs TH, Maekawa K, Obikawa T, Yamane Y (2000) Metal machining: theory and application. Edward Arnold, London, pp 35–80
12. Mok SL, Kwong CK, Lau WS (2001) A hybrid neural network and genetic algorithm approach to the determination of initial process parameters for injection molding. Adv Manuf Technol 18:404–409
13. Nakhjavani OB, Ghoreishi M (2006) Multi criteria optimization of laser percussion drilling process using artificial neural network model combined with genetic algorithm. Mater Manuf Process 21:11–18
14. António CC, Castro CF, Sousa LC (2004) Optimisation of metal forming processes. Comput Struct 82:1425–1433
15. Spears WM, DeJong SK (1991) On the virtues of parametrized uniform crossover. In: Belew RK, Booker LB (eds) Proceedings of Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, pp 230–236
16. Conceição António CA, Paulo Davim J (2002) Optimal cutting conditions in turning of particulate metal matrix composites based on experiment and a genetic search model. Compos Part A Appl Sci Manuf 33(2):213–219