Research Journal of Applied Sciences 1(1-4): 72-83, 2006

Recognition of Tool Wear by Using Extended Kalman Filter in Artifical Neural Network

[1]J. Hameed Hussain and [2]S. Purushothaman
[1]Department of Mechanical Engineering, Sathyabama Deemed University,
Old Mahabalipuram Road, Chennai, India-603103, India
[2]Department of Mechanical Engineering, Hindustan College of Engineering
Old Mahabalipuram Road, Chennai, India-603103, India

Abstract: The condition of the tool in a turning operation is monitored by using Artificial Neural Network (ANN). The recursive Kalman filter algorithm is used for weight updation of the ANN. To monitor the status of the tool, tool wear patterns are collected. The patterns are transformed from n-dimensional feature space to a lower dimensional space (two dimensions). This is done by using two discriminant vectors φ_1 and φ_2. These discriminant vectors are found by optimal discriminant plane method. Thirty patterns are used for training the ANN. A comparison between the classification performances of the ANN trained without reducing the dimensions of the input patterns and with reduced dimensions of the input patterns is done. The ANN trained with transformed tool wear patterns gives better results in terms of improved classification performance in fewer iterations, when compared with the results of the ANN trained without transforming the dimensions of the input patterns to a lower dimension.

Key words: Back-propagation algorithm, extended kalman filter, optimal discriminant plane

INTRODUCTION

In the manufacturing industries, automated machine tools are used. Some of them are single spindle, multispindle automats, capstan and turret and computer numerical control machines. In all these machines, predefined sequence of instructions, like using stops and programming methods, is used to execute the operations, so that good quality parts with mass production are achieved. When the tools are worn out, they are replaced with new tools, or reground and used. The duration, after which a tool has to be replaced or reground, can be expressed in terms of amount of flank wear land width of the tool (Vb) or tool life in minutes. Established data both in terms of tool life and amount of tool wear, are available, based on which, the tools can be replaced or reground. There is no assurance that the tool will last, till the established time. There is every possibility for the tool to fail in advance. Artificial neural network has been used to detect the amount of flank wear of the tool.

The methods, used for monitoring tool wear, are direct and indirect. The direct methods use measurements of volumetric loss of tool material. This procedure is done off-line. Some of the direct methods include change in work piece dimension, optical techniques, radioactive methods and pneumatic gauging method. The indirect methods use the measurement of cutting related parameters, like cutting forces, tool holder vibration, acoustic emission, etc. Due to the complexity and unpredictable nature of the machine process, the process has to be modeled with rule-based techniques. Modeling correlates process state variables to parameters. The process state variable is Vb. The process parameters are feed rate (F), cutting speed (S) and depth of cut (D_c). Some of the modeling techniques are multiple regression analysis and group method data handling. These methods require a relationship between process parameters and process state variables (Chryssolouris and Guillot[1-4].

The neural network approach does not require any modeling between process parameters and the outputs are process state variables. The network maps the input domains with the output domains. The inputs are process parameters and the outputs are process state variables. Each process parameter or process state variable is called feature. The combination of input and output constitutes a pattern. Many patterns will be called data.

In this study, instead of using the actual dimension of the input pattern (input vector), the dimension is reduced to two. The two dimensional input vector does not represent any individual feature of the original n-dimensional input pattern; instead, it is a combination of 'n' features of the original pattern. The

Corresponding Author: J. Hameed Hussain, Department of Mechanical Engineering, Sathyabama Deemed University, Old Mahabalipuram Road, Chennai, India-603103, India

components of the reduced pattern do not have any dimensional quantity.

Transformation of n-dimensional input patterns into two dimensional input vectors: The process of changing the dimensions of a vector is called transformation. The transformation of a set of n-dimensional real vectors onto a plane is called a mapping operation. The result of this operation is a planar display. The main advantage of the planar display is that the distribution of the original patterns of higher dimensions (more than two dimensions) can be seen on a two dimensional graph. The mapping operation can be linear or non-linear. Linear classification algorithm Fisher[5] and a method for constructing a classifier on the optimal discriminant plane, with minimum distance criterion for multiclass classification with small number of patterns Hong and Yang[6], have been developed.

The method of considering the number of patterns and feature size Foley[7] and the relations between discriminant analysis and multilayer perceptrons Gallinari[8], have been analyzed.

A linear mapping is used to map a n-dimensional vector space $\Re^n$ onto a two dimensional space. Some of the linear mapping algorithms are principal component mapping Kilter and Young[9]. generalized declustering mapping Sammon, Fehlauer, Gelsema and Eden[10-13], leased squared error mapping Mix and Jones[14] and projection pursuit mapping Friedman and Turkey[15].

In this study, the generalized declustering optimal discriminant plane is used. The mapping of the original pattern 'X' onto a new vector 'Y' on a plane is done by a matrix transformation, which is given by

$$Y=AX \tag{1}$$

where

$$A = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} \tag{2}$$

and φ_1 and φ_2 are the discriminant vectors (also called projection vectors).

An overview of different mapping techniques is given Siedlecki[16-17]. The vectors φ_1 and φ_2 are obtained by optimizing a given criterion. The plane formed by the discriminant vectors is the optimal vectors which are the optimal discriminant planes. This plane gives the highest possible classification for the new patterns.

The steps involved in the linear mappings are:

Step 1: Computation of the discriminant vectors φ_1 and φ_2: this is specific for a particular linear mapping algorithm.

Step 2: Computation of the planar images of the original data points: this is for all linear mapping algorithms.

Computation of discriminant vectors φ_1 and φ_2: The criterion to evaluate the classification performance is given by:

$$J(\varphi) = \frac{\varphi^T S_b \varphi}{\varphi^T S_W \varphi} \tag{3}$$

where,

S_b = the between class matrix and

S_w = the within class matrix which is non-singular.

$$S_b = \Sigma p(\omega_i)(m_i - m_o)(m_i - m_o)^T \tag{4}$$

$$S_w = \Sigma p(\omega_i) E\left[X_i - m_o)(X_i - m_i)^T \omega_i\right] \tag{5}$$

where

$P(\omega_i)$ a priori the probability of the ith pattern, generally, $p(\omega_i) = 1/m$

m_i the mean of each feature of the ith class patterns, (i=1.2…,m),

m_o the global mean of a feature of all the patterns in all the classes,

X = { x_i, I=1,2,…L} the n-dimensional patterns of each class,

L the total number of patterns.

Eq. 3 states that the distance between the class centers should be maximum. The discriminant vector φ_1 that maximizes 'J' in Eq. 3 is found as a solution of the eigenvalue problem given by :

$$S_b \varphi_1 = \lambda_{ml} S_w \varphi_1 \tag{6}$$

where:

λ_{ml} = the greatest non-zero eigenvalue of $(S_b S_w^{-1})$

φ_1 = eigenvalue corresponding to λml

The reason for choosing the eigenvector with maximum eigenvalue is that the Euclidean distance of this vector will be the maximum, when the compared with that of the other eigenvectors of Eq. 6. Another discriminant vector φ_2 is obtained, by using the same criterion of Eq. 3. The discriminant vector φ_2 should also satisfy the condition given by:

$$\varphi^T_2 \varphi_1 = 0 \quad (7)$$

Eq. 7 indicates that the solution obtained is geometrically independent and the vectors φ_1 and φ_2 are perpendicular to each other. Whenever the patterns are perpendicular to each other, it means, that there is absolutely no redundancy, or repetition of a pattern, during collection of tool wears patterns in turning operation. The discriminant vector φ_2 is found as a solution of the eigenvalue problem, which is given by:

$$Q_p \, S_b \, \varphi_2 = \lambda_{m2} \, S_w \, \varphi_2 \quad (8)$$

where

λ_{m2} the greatest non-zero eigen value of $Q_p \, S_b \, S_w^{-1}$ and
Q_p the projection matrix which is given by

$$Q_p = I - \frac{\varphi_1 \, \varphi_1^T S_W^{-1}}{\varphi_1^T \, S_W^{-1} \, \varphi_1} \quad (9)$$

where
I = an identity matrix

The eigenvector corresponding to the maximum eigenvalue of Eq. 8 is the discriminant vector φ_2.

In Eq. 6 and 8, S_W should be non-singular. The S_W matrix should be non-singular, even for a more general discriminating analysis and multiorthonormal vectors Foley and Sammon, Liu Cheng[18-20] If the determinant of S_W is zero, then Singular Value Decomposition (SVD) on S_W has to be done. On using SVD, S_W is decomposed into three matrices U, W and V. The matrices U and W are unitary matrices and V is a diagonal matrix with non-negative diagonal elements arranged in the decreasing order. A small value of 10^{-5} to 10^{-8} is to be added to the diagonal elements of V matrix, whose value is zero. This process is called perturbation. After perturbing the V matrix, the matrix S_w^1 is calculated by:

$$S_w^1 = U * W * V^T \quad (10)$$

where
S_W^1 the non-singular matrix which has to be considered in the place of S_w.

The perturbing value should be very minimum, which is just sufficient to make S_w^1 non-singular. The method of SVD computation and its applications are given[21-22]. As per Eq. 7, when the values of φ_1 and φ_2 are innerproducted, the resultant value should be zero. In reality, the innerproducted value will not be zero. This is due to floating point operations.

Computation of two-dimensional vector from the original n-dimensional input patterns: The two-dimensional vector set y_i is obtained by:

$$y_i = (u_I, v_I) = (X_i^T \varphi_1, X_i^T \varphi_2) \quad (11)$$

The vector set y_i is obtained by projecting the original pattern 'X' onto the space, spanned by φ_1 and φ_2 by using Eq. 11. The values of u_I and v_I can be plotted in a two-dimensional graph, to know the distribution of the original patterns.

3Basics of Artificial Neural Network (ANN): An artificial Neural Network (ANN) is an abstract simulation of a real nervous system that contains a collection of neuron units, communicating with each other via axon connections. Such a model bears a strong resemblance to axons and dendrites in a nervous system. Due to this self-organizing and adaptive nature, the model offers, potentially, a new parallel processing paradigm. This model could be more robust and user-friendly, than the traditional approaches. ANN has nodes or neurons, which are described by difference or differential equations. The nodes are interconnected layer-wise or intra-connected among themselves. Each node in the successive layer receives the inner product of synaptic weights, with the outputs of the nodes in the previous layer. The inner product is called the activation value. When the activation value is given as an input to a neuron, the output of the same neuron should lie in a closed interval [0,1]. To achieve this, sigmoid function is used to squash the activation value.

The supervised learning method is much suitable for learning tool wear where both inputs and outputs of the patterns are used for training the ANN. The commonly used supervised learning method is back-propagation algorithm BPA[23-26]. The main draw back of BPA is, that it is very slow in convergence and gets stuck up in the local minima[27] In order to increase the convergence rate, recursive extended Kalman filter algorithm is used.

Extended Kalman Filter algorithm (EKF): The algorithm uses a modified form of the BPA, to minimize the difference between the desired outputs and the actual outputs, with respect to the inner products to the non-linear function. But, in the conventional BPA, the

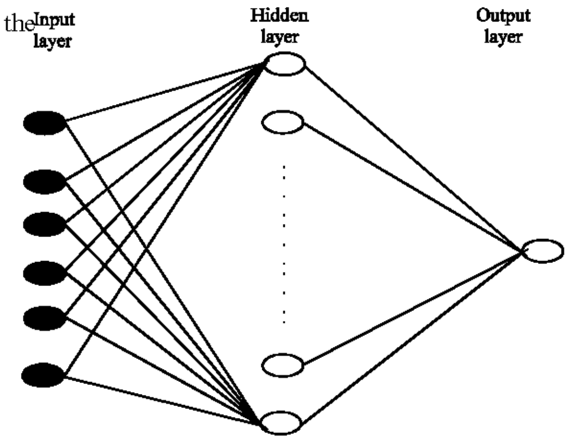

Fig. 1: Multilayer artificial neural network

difference between the desired outputs and the outputs of the network are minimized with respect to the weights. The EKF algorithm is a state estimation method for a non-linear system and it can be used as a parameter estimation method, by augmenting the state with unknown parameter[28]. A multi-layered network is a non-linear system with layered structure and its learning algorithm is regarded as parameter estimation for such a system[29-30]. The multi-layered ANN is shown in Fig. 1. The EKF based learning algorithm gives approximately the minimum variance estimates of the weights. The convergence of EKF is faster than that of BPA. Error values, which are generated by EKF, are used to estimate the inputs to the non-linearities. The estimated inputs, along with the input vectors to the respective nodes, are used to produce an updated set of weights, through a system of linear equations at each node. Using Kalman filter at each layer solves these systems of linear equations.

In EKF algorithm, the inputs to the non-linearities are estimated and its error co-variance matrix is minimized. This minimization of the co-variance of the vector helps in faster convergence of the network. The steps involved in training the ANN, by using extended Kalman filter algorithm are:

Step 1: Initialize the weights and thresholds randomly between layers, initial trace of the error co-variance matrix Q and the accelerating parameters λ and T_{max} to a very small value.

Step 2: Present the inputs of a pattern and compute outputs of nodes in the successive layers by:

$$\hat{X}_i^{n+1} = \frac{1}{1+\exp(-\sum W_{ij}\hat{X}_i + \Theta)} \begin{bmatrix} 1 \le i \le N_{n+1} \\ 1 \le n \le M-1 \end{bmatrix} \quad (12)$$

Step 3: Calculate the error E (p) of a pattern by:

$$E(p) = \frac{1}{2}\sum(d_i(p) - \hat{X}_i(p))^2 \quad (13)$$

and the Mean Squared Error (MSE) for all the patterns in iteration is obtained by:

$$E = \sum E(p) \quad (14)$$

where:

P the pattern number and

d the desired output.

Step 4: Calculate the accelerating parameter $\hat{\lambda}$ by:

$$\hat{\lambda}(p) = \hat{\lambda}(p-1) + \frac{1}{T_{max}}\left[\frac{(d(p)-\hat{X}(p))^T - (d(p)-\hat{X}(p))}{n_L} - \hat{\lambda}(p-1)\right] \quad (15)$$

Step 5: Allot output X of each node to $\hat{Y}$, which improves the estimation accuracy

$$\hat{Y}_i^{M-1}(p) = \hat{X}^M(p)\ [1 \le i \le N_M - 1] \quad (16)$$

For n=M-1 to 1 step-1

For I=1 to N_{n+1}

Step 6: Calculate the error δ at each node in the output layer by:

$$\delta_i^n(p) = \hat{X}_i^M (1 - \hat{X}_i^M) \quad (17)$$

Step 7: Calculate the temporary scalars β and α by:

$$\beta_i^n(p) = \delta_i^n(p)^T \delta_i^n(p) \quad (18)$$

$$\alpha_i^n(p) = \hat{X}_i^n(p)^T \psi_i^n(p) \quad (19)$$

where

$$\psi_i^n(p) = Q_i^n(p-1)\ \hat{X}^n(p) \quad (20)$$

Step 8: The weights W_{ij} are updated by:

$$W_{ij}^n(p) = W_{ij}^n(p-1) + \frac{\delta_i^n(p) R_i(p) \psi_i^n(p)}{\hat{\lambda}(p) + \alpha_i^n(p)\beta_i^n(p)} \quad (21)$$

where:

$$R_i(p) = d_i(p) - \hat{Y}_i^n(p) \text{ in the output layer} \quad (22)$$

$$R_i(p) = \hat{X}_i^n(p) - \hat{Y}_i^n(p) \text{ in the hidden layer} \quad (23)$$

Step 9: Update the error co-variance matrix Q by:

$$Q_i^n(p) = Q_i^n(p-1) + \frac{\beta_i^n(p)}{\hat{\lambda}(p) + \alpha_i \beta_i^n(p)} \psi_i^n(p) \psi_i^n(p)^T \quad (24)$$

Step 10: Update estimation accuracy $\hat{Y}$ in the hidden layers by:

$$\hat{Y}_{i+1}(p) = \hat{Y}_i^n(p) + \delta_i^n(p) \hat{X}_i^n(p)^T (W_{ij}^n(p) - W_{ij}^n(p-1)) \quad (25)$$
$$\hat{Y}_i^{n-1} = \hat{Y}_{N_{n+1}}^n$$

Step 11: Calculate the error δ at each node in the hidden layer by:

$$\delta_i^n(p) = \hat{X}_i^{n+1}(1 - \hat{X}_i^{n+1}) \sum \delta_j^n(p) W_{ij}(p) \quad (26)$$

Step 12: Adopt Eq. 14-26 until weights and thresholds between the layers are updated. Stop training the network once the performance index of the network is reached: otherwise continue with step 2.

Collection of data: Experimental study on turning was conducted with spheroidal graphite cast iron work material. The tool used for this work is made of 13 layers of coating with ALON, TiC, TiN, Ii(C, N) over a carbide substrate. This tool is named as Widalon HK15. The ranges of various process parameters are: cutting speed (200-500 m/min), feed (0.063-0.25 mm/rev) and depth of cut (0.5-2 mm). The turning operation was carried on a VDF high-speed precision lathe.

The three components of forces, namely, axial force (F_x), radial force (F_y) and tangential force (F_z) and flank wear land width (Vb) were collected. To read the cutting forces, a three-component piezoelectric crystal type of dynamometer (KISTLER type 9441) was used. The value of VB was measured by a toolmaker's microscope. The block diagram of the experimental set-up is shown in Fig. 2.

Experimental procedure: Turning was done for the combination of different speeds, feeds and depths of cut, using a fresh cutting edge. The ranges of cutting conditions were decided for progressive wear of the tool. For each cutting condition, the three components of cutting forces F_x, F_y and F_z, were measured.

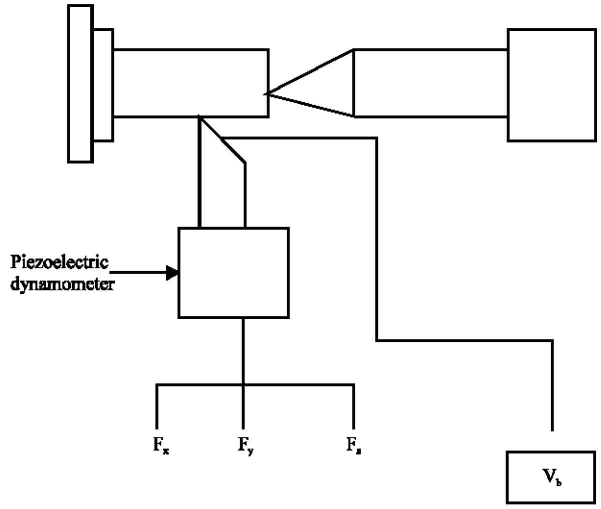

Fig. 2: Block diagram of the experimental setup

Measurements were made at different intervals of time. Depending on the length of cut, machining was stopped after every 60-80 seconds and Vb was measured. Static forces were recorded at two or three intermediate points between two wear measurements. The set of measurements, immediately prior to a wear measurement, had been used for training the neural network. About 113 patterns were collected. During re-insertion of tool inserts after every wear measurement, inserts were slugged into the slot made out in the tool holder, so that there was no change in the tool overhang.

Normalizing the patterns: The patterns are selected for training and testing. The inputs of the training and test patterns are normalized by:

$$X_{i=} X_i / X_1^2 + X_2^2 + \ldots. + X_n^2 \quad (27)$$

and the outputs are normalized by:

$$X_i = x_i / x_{max} \quad (28)$$

where:

x_I = the value of a feature and

x_{max} = the maximum value of the feature.

The reasons for using Eq. 27 to normalize the inputs of the patterns are:

- Each pattern is converted into unit length, so that the patterns lie in unit space,
- The vast difference among the values of features of a pattern is reduced,
- Eq. 4 and 5 represent autocorrelation technique. In such cases, the values of each feature of a pattern

Table 1: Patterns collected during turning

	Inputs							Outputs
S. No.	S (m/min)	F (mm/rev)	D_c (mm)	F_x (N)	F_y (N)	F_z (N)	Time (sec)	Vb (mm)
1	450	0.10	1.5	150	115	350	45	15
2	450	0.10	0.5	60	50	115	38	15
3	450	0.10	2.0	180	130	450	32	15
4	350	0.10	0.5	60	90	125	30	15
5	300	0.06	0.5	45	80	70	428	20
6	300	0.06	0.5	40	80	65	428	20
7	200	0.06	0.5	40	65	75	428	20
8	400	0.10	0.5	60	85	110	428	20
9	300	0.08	0.5	50	90	85	428	20
10	300	0.08	0.5	45	90	85	428	20
11	400	0.06	0.5	40	75	95	428	20
12	500	0.08	0.5	50	40	105	428	20
13	200	0.10	0.5	60	90	110	428	20
14	400	0.08	0.5	55	90	100	428	20
15	500	0.10	0.5	50	95	110	428	20
16	200	0.10	0.5	45	85	105	428	20
17	450	0.10	1.0	115	105	250	428	20
18	300	0.10	0.5	45	110	105	428	20
19	300	0.10	0.5	40	105	105	428	20
20	200	0.06	0.5	35	65	70	428	20
21	500	0.06	0.5	45	70	90	428	20
22	200	0.08	0.5	40	75	80	428	20
23	200	0.08	0.5	50	75	90	428	20
24	400	0.20	0.5	75	115	195	428	20
25	300	0.25	0.5	70	140	225	428	20
26	200	0.20	0.5	50	125	190	428	20
27	200	0.20	0.5	120	130	190	428	20
28	400	0.25	0.5	80	125	230	428	20
29	300	0.20	0.5	65	130	185	428	20
30	500	0.25	0.5	60	115	215	428	20
31	500	0.20	0.5	60	110	195	428	20
32	200	0.25	0.5	85	160	220	428	20
33	200	0.25	0.5	60	150	210	35	20
34	450	0.10	0.5	60	55	105	92	30
35	350	0.10	0.5	55	85	140	75	30
36	450	0.10	1.5	130	115	345	65	30
37	450	0.10	2	100	140	450	60	30
38	450	0.10	1	100	105	250	70	40
39	400	0.10	0.5	55	85	110	129	45
40	350	0.10	0.5	60	100	120	42	45
41	450	0.10	0.5	25	70	85	111	55
42	450	0.10	2	160	140	470	85	55
43	350	0.10	0.5	60	100	125	165	60
44	450	0.10	1.5	150	105	330	94	65
45	350	0.10	0.5	60	80	125	202	75
46	450	0.10	1	115	110	260	110	75
47	450	0.10	0.5	25	70	90	149	80
48	450	0.10	2	110	140	470	110	80
49	400	0.10	0.5	50	80	105	240	90
50	350	0.10	0.5	55	85	130	85	90
51	450	0.10	0.5	60	105	125	180	90
52	400	0.10	0.5	60	90	100	128	92
53	350	0.10	0.5	70	100	120	365	93
54	450	0.10	1.5	135	110	325	123	94
55	400	0.10	0.5	50	85	100	172	95
56	350	0.10	0.5	70	100	115	290	96
57	450	0.10	0.5	75	150	150	212	100
58	400	0.10	0.5	60	100	105	207	105
59	450	0.10	1	150	140	275	146	105
60	350	0.10	2	200	140	460	135	105
61	450	0.10	0.5	60	100	115	358	107
62	400	0.10	0.5	75	205	145	244	110

Continued

63	350	0.10	0.5	45	100	110	403	115
64	450	0.10	0.5	60	105	125	241	115
65	450	0.10	1.5	155	170	370	151	115
66	400	0.10	0.5	100	250	150	276	120
67	400	0.10	0.5	50	112	85	277	120
68	450	0.10	0.5	70	175	85	311	125
69	450	0.10	0.5	125	250	150	314	130
70	450	0.10	1.5	300	245	410	161	130
71	350	0.10	2	160	140	460	179	130
72	450	0.10	0.5	75	145	120	438	133
73	450	0.10	0.5	125	275	150	351	140
74	350	0.10	2	400	240	550	197	140
75	450	0.10	1.5	325	230	450	210	148
76	350	0.10	0.5	100	170	120	474	150
77	450	0.10	2	510	350	600	229	150
78	350	0.10	0.5	135	225	170	510	160
79	450	0.10	0.5	160	325	50	422	165
80	450	0.10	1.5	350	255	455	389	165
81	400	0.10	0.5	150	175	150	240	165
82	450	0.10	1.0	200	160	310	547	170
83	350	0.10	0.5	150	260	175	180	170
84	450	0.10	0.5	160	255	150	583	185
85	400	0.10	0.5	150	340	50	426	185
86	350	0.10	0.5	165	300	140	459	195
87	450	0.10	0.5	160	260	140	620	200
88	450	0.10	0.5	220	425	60	457	205
89	450	0.10	1.0	220	180	320	211	240
90	450	0.10	0.5	230	500	65	488	240
91	450	0.10	1.5	480	350	350	274	245
92	450	0.10	2.0	700	390	640	264	275
93	450	0.10	1.0	325	290	345	240	290
94	400	0.10	0.5	200	350	180	494	295
95	450	0.10	1.0	400	375	365	269	340
96	450	0.10	1.5	450	390	400	308	350
97	450	0.10	1.0	450	400	360	299	365
98	450	0.10	1.5	680	580	560	338	375
99	450	0.10	1.0	450	430	370	529	400
100	450	0.10	2.0	850	700	750	329	400
101	450	0.10	1.5	750	650	500	361	400
102	400	0.10	0.5	175	350	140	299	400
103	450	0.10	1.5	240	850	620	394	540
104	450	0.10	1.0	550	590	430	366	550
105	450	0.10	2.0	1100	1200	840	336	585
106	450	0.10	1.5	260	1200	800	427	690
107	450	0.10	1.0	570	700	450	403	755
108	450	0.10	2.0	1200	1400	1000	364	785
109	450	0.10	1.5	20	1800	1400	454	825
110	450	0.10	2.0	1500	1800	1000	396	880
111	450	0.10	1.0	950	700	500	440	980
112	450	0.10	1.5	380	1880	1500	482	980
113	450	0.10	2.0	1250	1440	1040	428	990

should be normalized within a close range of values and

- The number of floating point operations are minimized.

Selection of patterns for training ANN: The number of classes, the number of patterns in each class, the classification range in each class and the total number of training patterns are decided. If only one output is considered, the range of classification is simple. If more than one output is considered, a combination criterion has to be used. The remaining patterns which are used for training the network, should be, such that they represent the entire population of the data. The selection of patterns is done by:

$$E_i^2 = \frac{\sum_{j=1}^{nf}(x_{ij} - \bar{x}_j)^2}{\sigma_i^2} \qquad (29)$$

where

Table 2: Patterns used for finding out discriminant vectors φ_1 and φ_2 and training the ANN

Class	Pattern number
I	5,6,11,12,13,14,22,27,32,33,41,42,44,47,48,60,71,74,75,77
II	1.021051061 07108,109,110,111,112,113

Table 3: Number of patterns and classification range in each class

Pattern number and the class	No. of patterns in each class	No. of pattern used for training the ANN	No. of patterns used for testing the ANN	Range of V_b(mm)
1-87 Class I	87	20	67	<=200
88-113 Class 2	26	10	16	>200and<=990
Total	113	30	83	

E_i^2 the maximum variance of a pattern
Nf the number of features and

$$\sigma_i^2 = \frac{\sum_{j=1}^{nf}(x_{ij} - \bar{x}_j)^2}{L} \tag{30}$$

where

X_j the mean for each feature and
L the number of patterns.

The value of E_i^2 is found for the patterns given in Table 1. Patterns with maximum E_i^2 are chosen from each class for training the network and the corresponding pattern numbers are given in Table 2. The remaining patterns 83 are considered as test patterns. The classification ranges and the number of patterns used for training and testing the ANN, is given in Table 3.

Procedure for implementing optimal discriminant plane method in ANN: The steps involved in implementing optimal discriminate plane method in the ANN are as follows:

Step 1: Patterns for training the ANN are selected by using Eq. 29 and 30 and the remaining patterns are considered as test patterns.

Step 2: The inputs of each pattern in the training and test set are normalized, by using Eq. 27, so that the length of each pattern is one. The outputs of all the patterns are normalized by using Eq. 28.

Step 3: S_w and S_b matrices are calculated by using Eq. 4 and 5. The S_w matrix is checked for non-singularity. If S_w matrix is singular, singular value decomposition is applied to S_w and small perturbation is done. After perturbation, S_w matrix is recomputed as S_w^1.

Table 4: The 2-dimensional vectors of the normalized training patterns

Pattern No.	u	v
1	0.248405	0.113239
2	0.284535	0.146954
3	0.286177	0.157134
4	0.075246	-0.138888
5	0.126176	-0.220629
6	0.309048	0.021163
7	0.208151	-0.106131
8	0.377304	0.118302
9	0.033771	-0.257652
10	0.064219	-0.263593
11	0.427905	0.302125
12	0.296603	-0.172194
13	0.429281	0.293695
14	0.311602	-0.200051
15	0.344748	-0.018211
16	0.283635	-0.140848
17	0.296513	-0.163203
18	0.123165	-0.151501
19	0.136756	-0.104236
20	0.010996	-0.170528
21	-0.036130	-0.128443
22	-0.205433	-0.199332
23	-0.375849	-0.230142
24	-0.389305	-0.451422
25	-0.396734	-0.273041
26	-0.410809	-0.560982
27	-0.470258	-0.240555
28	-0.410351	-0.563859
29	-0.397719	-0.275753
30	-0.328632	-0.092794

Step 4: By using Eq. 6 and 8, the discriminant vectors φ_1 and φ_2 are calculated and are given by

$$\varphi_1 = \begin{bmatrix} +0.535115 \\ +0.000019 \\ +0.002912 \\ -0.143451 \\ -0.805960 \\ +0.208547 \end{bmatrix}, \varphi_2 = \begin{bmatrix} +0.483875 \\ -0.000229 \\ -0.000692 \\ +0.341656 \\ -0.347608 \\ -0.726844 \end{bmatrix} \tag{31}$$

Step 5: The normalized inputs of the training patterns are transformed into two-dimensional vectors by using Eq. 3. The two-dimensional vectors for the training patterns are given in Table 4.

Training the ANN Off-line

Step 6: The normalized inputs of the test patterns are transformed to two-dimensional vector, by using Eq. 31.

Step 7: The two dimensional vectors of the training patterns are used as the inputs and the corresponding normalized outputs to the ANN. Presenting all the training vectors forms one iteration.

Step 8: At the end of each iteration, the two dimensional vectors of the test patterns are presented to the network.

If the classification performance of the ANN is not satisfactory, step 7 is adopted. Otherwise, training of the ANN is stopped and the weights obtained in the last iteration is considered, as the final weights.

Implementing the ANN online for tool wear monitoring: Based on the above, an on-line method for tool wear condition monitoring is suggested below:

Step 1: The final weights of the network, obtained during training, is stored in the database.

Step 2: The cutting forces are collected from the dynamometer and speed, feed and depth of cut are given as the inputs to the ANN.

Step 3: The inputs are normalized and transformed into two-dimensional vectors by using the φ_1 and φ_2 discriminant vectors.

Step 4: Classification rules are written to check the output of the network. If the output of the network is within the specified value, step 2 is continued. Otherwise, corrective actions are implemented. Some of the corrective actions are:

- Stopping the machining operation, or
- Regrinding the tool, or
- Replacing the worn out tool with a new tool using the automatic tool changer (ATC).

RESULTS AND DISCUSSION

The transformed vectors given in Table 4 are shown in Fig. 3. The x-axis represents values of ‘u’ and y-axis represents values of ‘v’. As per Table 3, 20 patterns from class 1 and 10 patterns from class 11 have been used for training the ANN. It can be observed from Table 1 that most of the patterns are in class 1 and hence more number of patterns have been chosen from class 1, when compared with that of patterns chosen from class 11. The legend ‘X’ represents 20 patterns from class 1 and clearly forms one group and legend ‘Δ’ represents 10 patterns from class 11 and forms a separate group. There is no mixing of patterns belonging to one class with another class. Eq. 3 has ensured that the class I and class 11 are separate. The transformation method also helps us in visualizing of how the given set of patterns is distributed in space.

The ANN is trained with different number of nodes in the hidden layer, to find out the exact number of hidden nodes required to represent the tool wear data. The exact

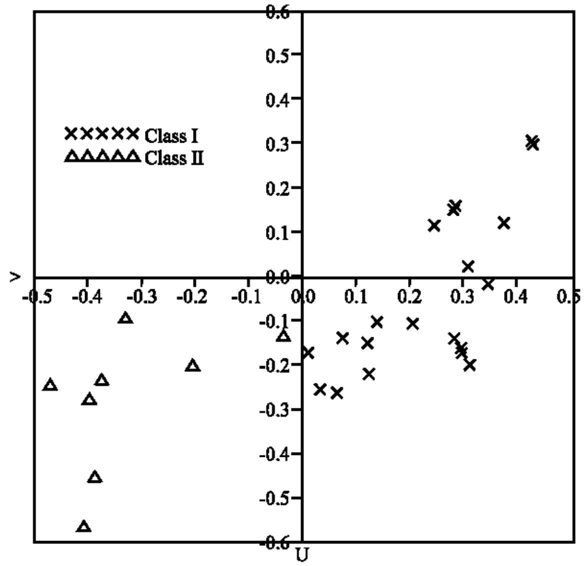

Fig. 3: Distribution of the 30 training patterns

number of nodes obtained is 6. Only one hidden layer is used. Using more than one hidden layer will only increase the number of arithmetic operations. Normally, one hidden layer is sufficient.for the ANN to represent most of the patterns available. Thresholds are not used during training, as they increase the number of iterations to reach the desired performance index of the network. The range of initial weights used is 0.25-0.45. The classification performance of the network has been taken as criteria to stop the training of the network. During training of the network, at the end of each iteration, all the test patterns are presented to the network and their correct classification is noted. While presenting the test patterns to the network, weight updating is not done. If the classification performance is not up to the expectation, training of the network is continued. Once the desired classification performance is obtained, training of the network is stopped: and the weights are treated as the final weights for on-line implementation.

Training the ANN by using EKF without reducing the dimensions of the inputs of the tool wear patterns from 6 to 2: The network is trained by using EKF weight updating algorithm to learn the tool wear data. The inputs of the training patterns are presented to the network without reducing their dimensions. The training conditions used are: momentum factor as 0.5, initial value for the error co-variance matrix Q as 20 and the initial value for the accelerating parameter T_{max} as 20. The number of nodes in the hidden layer is 6. The configuration of the network is 6-6-1. The above values for Q and T_{max} are obtained by simulation of network. A

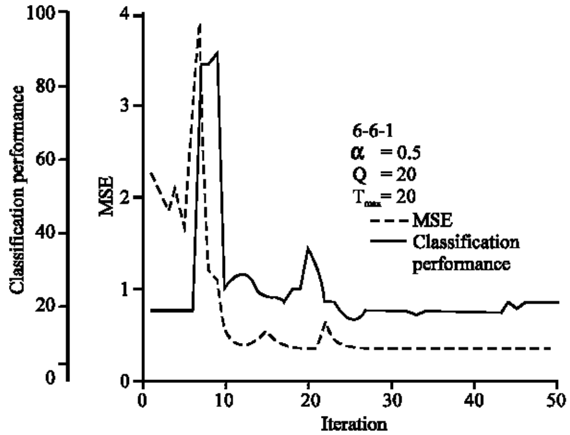

Fig. 4: MSE and classification performance of the network trained by using EKF without transforming the inputs of the tool wear patterns

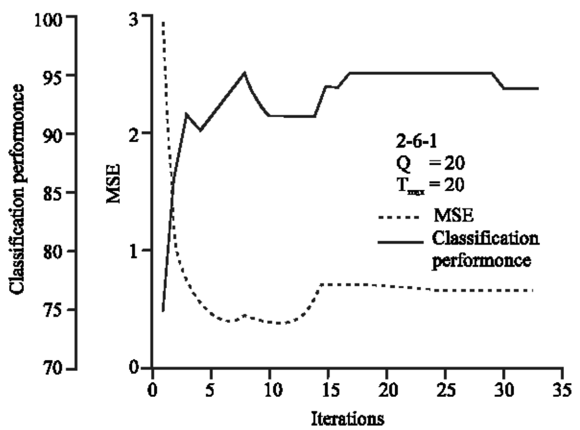

Fig. 5: MSE and Classification performance of the network trained by using EKF with transformed inputs of the tool wear patterns

Table 5: Comparison of the Performance of the net work trained by using EKF with and with out transformed input patterns

	Configuration	Classification performance in percentage	Iterations	MSE
With out Transforming input patterns	6-6-1	89.16	9	1.1277
with transformed input patterns	2-6-1	95.18	8	0.4666

maximum classification performance of 89.15% is obtained in 9 iterations. The Mean Squared Error (MSE) is 1.12779. The MSE is the summation of the square of the difference between the desired and actual outputs of the network for all the training patterns. The classification performance and MSE curves are shown in Fig. 4. The classification performance increases up to the 9th iteration and then decreases. There is no improvement in the classification performance during further training of the network.

Training the ANN by using EKF by reducing the dimensions of the inputs of the tool wear patterns from 6 to 2: The network is trained by using EKF. The inputs of the training patterns are transformed into 2 dimensions and presented to the network. The initial value for the accelerating parameter T_{max} is 20 and initial value for the error co-variance matrix Q is 20. The configuration of the network is 2-6-1. A maximum classification performance of 95.18% is obtained in 8 iterations at MSE of 0.4666 (Fig. 5), whereas the classification performance of the network is only 89.15% (Fig. 4) when trained without transforming the dimensions of the input patterns. Because of the transformation of the input patterns the size of the network is reduced, the classification performance has increased and the iterations, at which maximum classification performance is obtained, are reduced. A comparison of the performance of the network trained by using EKF with transformed input patterns and without transforming the input patterns is given in Table 5. From Table 5, it can be seen that the network trained with transformed input patterns gives higher classification performance.

CONCLUSION

The network trained with transformed input patterns has multifold advantages than the network trained without transforming the input patterns. The advantages are the number of nodes in the input layer is reduced and hence the size of the network also is reduced the number of arithmetic operations is drastically reduced and the iterations, at which maximum classification performance is obtained, are reduced.

Nomenclature:

ANN	artificial neural network
BPA	back-propagation algorithm
di (P)	desired output of the pattern 'p'
D_c	depth of cut, mm
EKF	extended Kalman filter
F	feed, mm/rev
F_x	axial force, N
F_y	radial force, N
F_z	tangential force, N
J (Ô)	objective function in Fisher's criterion
L	total number of patterns
M	total no of layers including the input and output layers
m_i	mean of each feature of the ith class patterns
m_o	global mean feature of all the patterns in all the classes

n, i	indices
N_n	total number of nodes in the nth layer
P	pattern number
Q	error co-variance matrix
Q_p	projection matrix
S	speed m/min
S_b	between class matrix
S_w	within class matrix which is non-singular
SVD	singular value decomposition
Vb	flank wear land width of turning tool μm
$W_{i,j}$	weights between layers
X	n-dimensional patterns of each class
x_i	value of a feature and also the outputs of neurons in the hidden and output layers
Xmax	maximum value of a feature
X_i^n (p)	out of the node, 'i' in the nth layer for pattern 'p'
	αmomentum or accelerating factor
	δerror of the nodes in the hidden and output layers learning factor
θ_i	thresholds of the nodes
y	estimation accuracy parameter
λ, T_{max}	accelerating parameters
λ_{m1}, λ_{m2}	greatest non-zero eigenvalue
φ_1, φ_2	projection vectors or discriminant vectors

REFERENCES

1. Chryssolouris, G. and M. Guillot, 1988. An A.I. approach to the selection of process parameters in intelligent machining, The winter annual meeting of the ASME, Chicago, Illinois.
2. Lee, D., I. Hwang and D. Dornfeld, 2004. Precision manufacturing process monitoring with acoustic. emission, proceedings AC'04, Zakopane, Poland.
3. Li Xiaoli and Xin Yao, 2005. Multi-scale statistical process monitoring in machining, IEEE Trans On industrial electronics, 52: 924-926.
4. Tug, O., Orul Zel and Yigoit Karpat, 2005. Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks, Intl. J. Machine Tools and Manufacture, 45: 467-479.
5. Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems, Ann Of Eugenics, pp: 178-188.
6. Hong, Z.Q. and Y.J. Yang, 1991. Optimal discriminate plane for a small number of samples and design method of classifier on the plane, pattern recognition, pp: 317-324.
7. Foley, D.H., 1972. Consideration of sample and feature size, IEEE Trans., on Inf. Theory, pp: 626-681.
8. Gallinari, P., S. Thiria, F.F. Badran and F. Fogelman-Soulie, 1991. On the relations between discriminant analysis and multilayer perceptrons, Neural Networks, pp: 349-360.
9. Kittler, J., P.C. Young, 1973. Approach to feature selection based on the Karhunen-Loeve expansion, Pattern Recognition, pp: 335-352.
10. Sammon, J.W., 1970. An optimal discriminant plane, IEEE Trans.. On Comp., pp: 826-829.
11. Sammon, J.W., 1970. Interactive pattern analysis and classification, IEEE Trans. On Comp., pp: 594-616.
12. Fehlauer, J. and B.A. Eisenstein, 1978. A declustering criterion for feature extraction in pattern recognition, IEEE Tran. On Comp., pp: 261-266.
13. Gelsema, E. and R. Eden, 1980. Mapping algorithms in ISPAHAN, pattern Recognition, pp: 127-136.
14. Mix, D.F. and R.A. Jones, 1982. A dimensionality reduction techniques based on a last squared error criterion, IEEE trans.. Patt. Ana. Mach. Intl., pp: 537-544.
15. Friedman, H. and J.W. Turkey, 1974. A projection pursuit algorithm for exploratory data analysis, IEEE Trans.. On Comp., pp: 881-890.
16. Siedlecki, W., K. Siedlecka and J. Skalansky, 1988. An overview of mapping techniques for exploratory data analysis, Pattern Recognition, pp: 411-429.
17. Siedlecki, W., K. Siedlecka and J. Skalancky, 1988. Experiments on mapping techniques for exploratory Pattern analysis, Pattern Recognition, pp: 431-438.
18. Foley, D.H. and J.E. Sammon, 1975.An optimal set of dicriminant vectors, IEEE Trans. On Comp., pp: 281-289.
19. Liu, K., Y.Q. Cheng and J.Y. Yang, 1992. A generalized optimal set of discriminant vectors, Pattern Recognition, pp: 731-739.
20. Cheng, Y.Q., Y.M. Zhuang and J.Y. Yang, 1992. Optimal fisher discriminant analysis using the rank decomposition, Pattern Recognition, pp: 101-111.
21. Klema, V.C. and A.J. Laub, 1980. The Singular value decomposition: its computation and some applications, IEEE Trans. on Automatic Control, 25: 164-176.
22. Sullivan, B.J. and B. Liu, 1984. On the use of singular value decomposition and decimation in discrete-time band-limited signal extrapolation, IEEE Trans.. On acoustics, speech and signal processing, pp: 1201-1212.
23. Lippmann, R.P., 1987. An introduction to computing with neural nets, I IEEE Trans. On Acoustics, speech and signal processing magazine, pp: 4-22.
24. Rumelhart, D.E., G.E. Hinton and R.J. William, 1986. Learning internal representations by error propagation, in D.E.Rumelhart, J.L.McClelland., Parallel distribution processing: Explorations in the microstructure of cognition, Foundations, M.I.T., Press, Cambridge, MA.

25. Widrow, B. and M.E. Hoff, 1960. Adaptive switching circuits, 1960 IRE WESSCON Convention Record, New York: IRE, pp: 96-104.
26. Hush, D.R. and B.G. Horne, 1993. Progress in supervised neural networks, IEEE Signal Proc. Mag., pp: 8-12.
27. Shih-Chi Huang and Yih-Fang Huang, 1990. Learning algorithms for perceptrons using back propagation with selective updates, IEEE Cont. Systems Mag., pp: 56-61.
28. Kalman, R.E. and R.S. Bucy, 1961. New result in linear filtering and prediction theory, Tran. Of the ASME-J. Of Basic Eng., pp: 95-108.
29. Liguni., Y., H. Sakai and H. Tokumaru, 1992. A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter, IEEE Trans. On Acoustics, Speech and Signal Processing, pp: 959-966.
30. Scalero, R.S. and N. Tepedelenliegu, 1992. A fast new algorithm for training feed forward neural networks, IEEE Trans. On Acoustics, speech and signal Processing, pp: 202-210.