THE MONITORING OF A LATHE USING AN ARTIFICIAL NEURAL NETWORK – 6th PART (STATISTICS, MONITORING, FUZZY C-MEANS METHOD)*

George C. BALAN, Alexandru EPUREANU, Ciprian CUZMIN

The study of machine-tool dynamic is realized here as "monitoring", meaning checking and improving the functioning of the machine. The state of processing is followed by certain sensors whose signs are processed inside the computer and then it takes the decision of monitoring, meaning the identification of a class from the set of classes (process conditions). In this part of the paper there are presented statistics, monitoring and fuzzy c-means method.

1. INTRODUCTION

In the first part of this paper there are presented the classes (tool conditions) for monitoring in turning and the artificial neural networks – **ANN**s (the creation of an ANN with the function *newff*).

In the second part of this paper there are presented the batch training of an artificial neural network with Levenberg-Marquardt algorithm and the experimental setup for components of cutting forces.

In the third part there are presented the experimental setup (for: cutter holder accelerations, cutting temperature, surface roughness, power), cutting working conditions and the tool wear.

In the fourth part there are presented the experimental results (validation of recordings) and the data processing (initial processing).

In the fifth part of the paper there are presented recordings spectral analysis and use of ANN on monitoring of the tool wear.

In this part of the paper there are presented statistics, monitoring and fuzzy c-means method.

2. STATISTICS

In the input matrix p, having sizes 12 (monitoring indices) × 750 (recordings), for each column (recording), we calculate:

– the average of the 12 indices μ;

---

* Presented at SISOM 2007 and Homagial Session of the Commision of Acoustics, Bucharest, 29–31 May.

"Dunărea de Jos" University of Galați, Str. Domnească nr. 47, Galați – 800 008, Romania, E-mail: George.BALAN@ugal.ro

Rev. Roum. Sci. Techn. – Méc. Appl., TOME 53, N° 2, P. 157–164, Bucarest, 2008

– the standard deviation $\sigma = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(x_k - \mu)^2}$, where $N = 12$;

Figure 1 presents the results, in which the recordings 1 ÷ 160, $c_2 \rightarrow$ 161 ÷ ÷265, $c_3 \rightarrow$ 266 ÷ 485, $c_4 \rightarrow$ 486 ÷ 545, $c_5 \rightarrow$ 546 ÷ 605, $c_6 \rightarrow$ 606 ÷ 675, $c_7 \rightarrow$ 676÷ ÷ 750 are allocated to class c_1.

The two graphs look alike, and the classes are easy to separate (depending on the values of μ and σ). We conclude that monitoring indices were selected correctly.

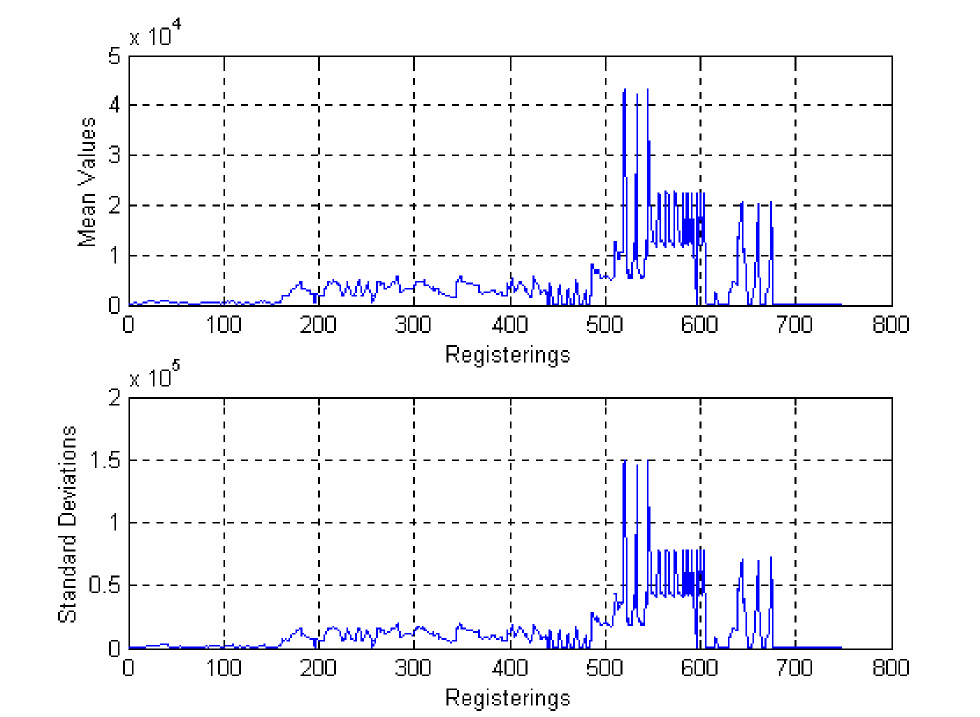

Fig. 1

3. MONITORING

Monitoring the tool wear involves that during the **continuous** process of cutting (Fig. 2) the "Spider" device should connected into the system, and it should transmit a recording to the PC, based on which ANN will say the class the processing is into (in conformity with [2], table 2.2). Then "Spider" connects itself again, a. s. o. In case the class exceeds 3 (therefore "abnormal"), PC will produce a sonorous signal, or stop the processing.

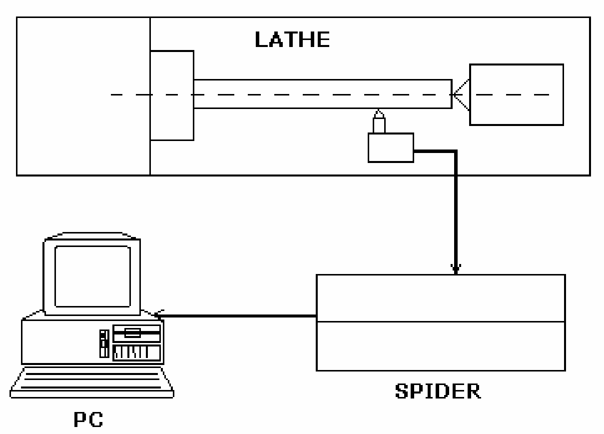

Fig. 2

To detail: the recording has 4800 samples, being a table in EXCEL with 4800 rows and 3 columns ($A = \varepsilon_1^{inreg}$, $B = \varepsilon_2^{inreg}$, $C = a_z$). The transfer Spider → PC is carried out within nearly 1 min. Out of this table a new (smaller) table is selected and it will consists of 960 rows and 4 columns, the first element being selected at random, at a location higher than A500. Function F_z (= $1136 \times A – 1136 \times B$) is calculated in fourth column, according to [3], formula (15). This table is transferred in MATLAB, where 12 monitoring indices will be calculated and then they are presented at the ANN input, as shown in the final section of chapter 3 from [6].

To see how ANN responds we use some of the initial recordings (described at the end of chapter 6.1); the results are presented in Table 1.

Table 1

No	Recor	Class	*v* [m/min]	*t* [mm]	*s* [mm/rot]	Epochs	Error %	Class from RNA	Time [min]
0	1	2	3	4	5	6	7	8	9
1	046	1	69	0.5	0.25	43	4.93	1	10
2	183	5	71	3	0.302	40	4.4	5	8
3	168	4	75.4	2	0.302	76	2.5	4	6
4	174	4	120.6	2	0.416	67	3.47	4	8
5	099	2	83.2	1	0.334	43	4.93	2	
6	190	7	0	0	0	47	3.6	7	7
7	095	6	83.2	1	0.353	47	23.87	6	7

With the first 4 recordings, after the first run a second run was carried out, and it lasted 2 min, the class given by ANN coinciding with the one of the first run.

The coincidence of the values in columns 2 and 8 shows that ANN provides correct outputs.

To asses the soft efficiency, the runs will be resumed, as in table 1, but for one third of the 44 recordings of class 3. Table 2 presents the results; *L* is the distance from the centre of the cut zone to universal. It is noticed that only the recordings under no. 1 and no. 5 did not give the correct class (3), but a neighbouring class. Other two runs carried out for each of these recordings – with the modification of the location of the first element in the table (960 × 4) – showed the correct class (3). Therefore, with 25 de runs only two were erroneous, the error amounting to 8 %. We consider this error to decrease in the future if we take several recordings in Tables 1 and 2.

Table 2

No.	Recor.	*v* [m/min]	*t* [mm]	*s* [mm/rot]	*L* [mm]	Epochs	Error %	Class in RNA
0	1	2	3	4	5	6	7	8
1	115	130.6	1.2	0.334	245	40	4.4	**2**
2	118	130.6	1.2	0.416	90	76	2.53	3
3	126	79.3	1.4	0.292	770	43	4.93	3
4	129	79.3	1.4	0.353	630	43	4.93	3
5	133	63.4	1.4	0.353	425	40	4.4	**4**
6	136	63.4	1.4	0.292	300	76	2.53	3
7	139	63.4	1.4	0.212	200	64	2.67	3
8	142	63.4	1.4	0.167	120	67	3.47	3
9	146	77.3	1.5	0.375	780	43	4.93	3
10	149	77.3	1.5	0.302	615	47	3.6	3
11	152	77.3	1.5	0.25	480	47	23.87	3
12	155	123.7	1.5	0.177	340	40	4.4	3
13	158	123.7	1.5	0.146	230	31	15.2	3
14	161	123.7	1.5	0.118	130	76	2.53	3

4. FUZZY C-MEANS METHOD

To compare with ANN method, will shall classify the 750 recordings making use of *Fuzzy c-means method* [7, 9].

Fuzzy *clustering i*s a technique used to determine the optimal clusters in a space of vectors, based on **Euclidean norms** for distance between the vectors.

Fuzzy c-means (FCM) is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade. It provides a method of how to group data points that populate some multidimensional space into a specific number of different clusters.

For the set of data from [2], Table 1:

$$X = (\boldsymbol{x}_1, \boldsymbol{x}_2\,, \ldots, \boldsymbol{x}_N), \tag{1}$$

the objective is to classify the data into n fuzzy clusters (classes). A membership value which describes the degree of belonging of each datum to each class is:

$$\mu_{\mathrm{k,\,j}}\;(\mathbf{x}_{\mathrm{k}}) \in [0,\,1],\;\; (1 \le j \le n,\;\; 1 \le k \le N). \tag{2}$$

The *partition* matrix is defined as follows:

$$\mu(n) = \begin{bmatrix} \mu_{11}(x_1) & . & . & \mu_{1N}(x_N) \\ . & & & . \\ . & & & . \\ \mu_{n1}(x_1) & . & . & \mu_{nN}(x_N) \end{bmatrix}, \tag{3}$$

where the membership values are subject to the following conditions:

$$\sum_{j=1}^{n} \mu_{kj} = 1 \,,\ (k = 1, \ldots, N)\,. \tag{4}$$

The performance index for the partition matrix is given by:

$$J = \sum_{k=1}^{N} \sum_{j=1}^{n} \ (\mu_{kj})^{w} \, \|\boldsymbol{x}_k - \boldsymbol{v}_j\|^{2}, \tag{5}$$

where $\boldsymbol{v}_j$ is a pseudo-centre of gravity vector of the j-th class:

$$\boldsymbol{v}_j = \frac{\sum_{k=1}^{N} (\mu_{kj})^{w} \cdot x_k}{\sum_{k=1}^{N} (\mu_{kj})^{w}}, (j = 1\,,\, 2\,,\, .\,.\,.\,,\, n) \tag{6}$$

and $w > 1$ is a weight of membership values; $\| \boldsymbol{x}_k - \boldsymbol{v}_j \|$ is an Euclidean norm used as the distance between $\boldsymbol{x}_k$ and $\boldsymbol{v}_j$:

$$\| \boldsymbol{x}_k - \boldsymbol{v}_j \| = \sqrt{\sum_{i=1}^{m} (x(k,i) - v(j,i))^2}\ . \tag{7}$$

When the performance index J attains its minimum, which indicates that the sum of the within class fluctuations has the minimum value, the *optimal* partition matrix is obtained.

In the learning phase, the procedure is:

1 – Construct the partition matrix as follows. Heuristically determine $\mu^{t}_{k\,,\,j}$, which satisfy (4). Then:

$$\mu(n) = \begin{bmatrix} \mu^{t}_{11}(x_1) & . & . & \mu^{t}_{1N}(x_N) \\ . & & & . \\ . & & & . \\ \mu^{t}_{n1}(x_1) & . & . & \mu^{t}_{nN}(x_N) \end{bmatrix}, \tag{8}$$

Go to step 2, with the initial value $t = 0$.

2 – Calculate the pseudo-centres of gravity vector of each cluster for the partition matrix:

$$\boldsymbol{v}_j^t = \frac{\sum_{k=1}^{N} (\mu_{kj}^t)^w \cdot x_k}{\sum_{k=1}^{N} (\mu_{kj}^t)^w}, \; (j = 1, \ldots, n). \tag{9}$$

3 – Modify each membership value as follows:

$$\mu_{kj}^{t+1}(x_k) = \frac{1}{\sum_{\alpha=1}^{n} \left(\frac{\left\| x_k - v_j^t \right\|}{\left\| x_k - v_\alpha^t \right\|} \right)^{1/(w-1)}}, \text{ if } \boldsymbol{x}_k \neq \boldsymbol{v}_j^t; \tag{10}$$

while for $\boldsymbol{x}_k = \boldsymbol{v}_j^t$, then: $\mu^{t+1}{}_{k,j}(\boldsymbol{x}_k) = 1$, if $j = \alpha$; $\mu^{t+1}{}_{k,j}(\boldsymbol{x}_k) = 0$, if $j \neq \alpha$. The modified partition matrix is re-organized:

$$\mu^{t+1}(n) = \begin{bmatrix} \mu_{11}^{t+1}(x_1) & . & . & \mu_{1N}^{t+1}(x_N) \\ . & & & . \\ . & & & . \\ \mu_{n1}^{t+1}(x_1) & . & . & \mu_{nN}^{t+1}(x_N) \end{bmatrix}. \tag{11}$$

4 – Determine if the convergence of the partition matrix is achieved:

$$\left| \mu^{t+1}{}_{kj} - \mu^{t}{}_{kj} \right| < \varepsilon, \tag{12}$$

where ε is a threshold of convergence. If the convergence isn't achieved, go back to the step 2 with $t = t + 1$.

In the *classification phase*, based on the *new* sample $\boldsymbol{x}$, and the cluster centres [the final expressions from (9)], on can calculate the fuzzy degree of the new sample using a relation like (10):

$$\mu_{kj}(x) = \frac{1}{\sum_{\alpha=1}^{n} \left(\frac{\left\| x - v_j \right\|}{\left\| x - v_\alpha \right\|} \right)^{1/(w-1)}}, \text{ if } \boldsymbol{x} \neq \boldsymbol{v}_j; \tag{13}$$

while for $\boldsymbol{x} = \boldsymbol{v}_j$, then: $\mu_j(\boldsymbol{x}) = 1$, if $j = \alpha$ and $\mu_j(\boldsymbol{x}) = 0$, if $j \neq \alpha$.

Accordingly, the process conditions is identified by the maximum fuzzy degree.

In MATLAB, the Fuzzy Logic Toolbox command line function *fcm* starts with an initial guess for the cluster centres, which are intended to mark the average location of each cluster. The initial guess for these cluster centres is most likely incorrect. Additionally, *fcm* assigns every data point a membership grade for each cluster. By iteratively updating the cluster centres and the membership grades for each data point, *fcm* iteratively moves the cluster centres to the "right" location within a data set.

For input argument "data" equal with the matrix $p'= x(750\times12)$ in chapter 2 (on a column in U (7×750) being the membership degrees of a vector to the seven classes), the programme which attributes the vector of the class for which the membership degree has the maximum value, will reveal the result in Fig. 3 and below.

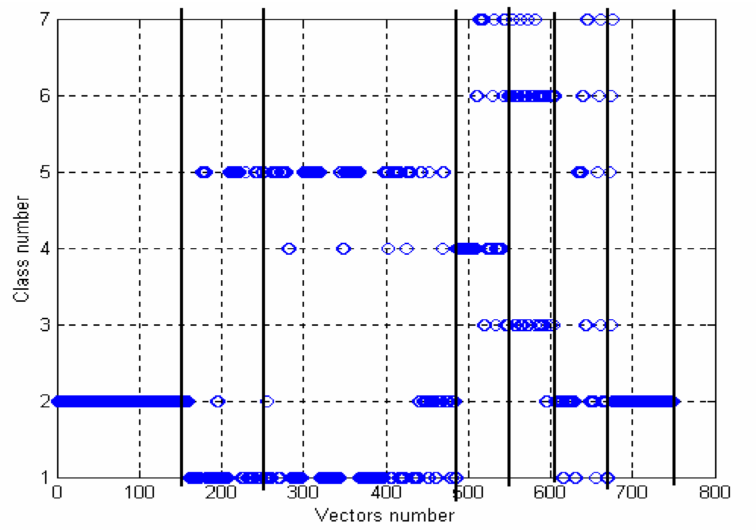

Fig. 3

Iteration count = 1, obj. fcn = 647350640000.421870
. . .
Iteration count = 31, obj. fcn = 121863345485.359880
. . .
Iteration count = 61, obj. fcn = 120594760756.125920
. . .
Iteration count = 91, obj. fcn = 120594676310.747500
. . .
Iteration count = 108, obj. fcn = 120594676310.453870
Iteration count = 109, obj. fcn = 120594676310.453890
Iteration count = 110, obj. fcn = 120594676310.453890
Column no. 100 in U: U100 = [0.0006 0.9990 0.0000 0.0001 0.0002 0.0000 0.0000]^T

A second run results in an image similar to Fig. 3, the horizontals having the same components, but on other levels.

A similar programme, which used only 4 columns of matrix $x(750\times12)$ namely the most important ones: 4 (F_z), 6 (Number of intersections), 7 (F_z spectral density) and 10 (a_z spectral density) give the same results.

In addition, the same results were obtained and when data set to be clustered had two columns: the values in Fig. 1.

Figure 3 is interpreted as follows: classes established under ANN method [6, Ch. 3.2] are partial found in Fig. 3, being numbered in a different order (c_1 is now class 2, and c_7 is still 2, which is not correct; c_2 is 1, the majority of recordings being on level 1; c_3 is divided on levels 1 and 5, which is not acceptable; c_4 is 4, the majority of recordings being on level 4, etc.). Few recordings have been noticed to be allocated to three classes (3, 4 si 7). In conclusions, fuzzy c-means method produces many more errors than the ANN method.

Acknowledgements. This research was supported by a grant CNCSIS [1] and a contract CEEX [6].

REFERENCES

1. BALAN, G., *Monitorizarea unui strung utilizand o retea neuronala artificiala*, Contract (grant tip A) nr. 33 445 / '02, Tema 19, Cod CNCSIS 451.
2. BALAN, G., EPUREANU, A., *The monitoring of a lathe using an artificial neural network – 1-st part (Theoretical basis)*, The Annual Symposium of the Institute of Solid Mechanics (SISOM 2004), Edit. Academiei Române, Bucharest, 2005.
3. BALAN, C. G., EPUREANU, A., POPESCU, F., *The monitoring of a lathe using an artificial neural network – 2nd part (the training of an artificial neural network and the experimental setup)*, The Annual Symposium of the Institute of Solid Mechanics (SISOM 2005), Edit. Academiei Române, Bucharest, 2006.
4. BALAN, C. G., EPUREANU, A., VACARUS, V., *The monitoring of a lathe using an artificial neural network – 3rd part (the experimental setup)*, The Annual Symposium of the Institute of Solid Mechanics (SISOM 2006), Edit. Academiei Române, Bucharest, 2007.
5. BALAN, C. G., *The monitoring of a lathe using an artificial neural network – 4th part (experimental results, data processing)*, The Annual Symposium of the Institute of Solid Mechanics (SISOM 2006), Edit. Academiei Române, Bucharest, CD-R.
6. BALAN, C. G., EPUREANU, A., CUZMIN, G., *The monitoring of a lathe using an artificial neural network – 5th part (recordings spectral analysis, use of ANN on monitoring of the tool wear)*, The Annual Symposium of the Institute of Solid Mechanics (SISOM 2007), Edit. Academiei Române, Bucharest, CD-R.
7. BEZDEK, J.C., 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.
8. EPUREANU, A., 2005, *Simulation, modeling and virtual production methods based on informatics and communication technologies*, Contract no. 22 CEEX I03/'05, MEdC.
9. RYAN, J., RYAN, M., POWER, J., 1994, *Using fuzzy logic*, Prentice Hall.