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 – ANNs (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) $\times$ 750 (recordings), for each column (recording), we calculate:

– the average of the 12 indices $\mu$ ;

---


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

- 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 \( \div \) 160, \( c_2 \rightarrow 161 \div +265, c_3 \rightarrow 266 \div 485, c_4 \rightarrow 486 \div 545, c_5 \rightarrow 546 \div 605, c_6 \rightarrow 606 \div 675, c_7 \rightarrow 676 \div 750 \) are allocated to class \( c_1 \).

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

![Figure 1](image)

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.
To detail: the recording has 4800 samples, being a table in EXCEL with 4800 rows and 3 columns \((A = \varepsilon_1^{\text{inreg}}, B = \varepsilon_2^{\text{inreg}}, C = a_z)\). 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>
<thead>
<tr>
<th>No</th>
<th>Recor</th>
<th>Class</th>
<th>(v) [m/min]</th>
<th>(t) [mm]</th>
<th>(s) [mm/rot]</th>
<th>Epochs</th>
<th>Error %</th>
<th>Class from RNA</th>
<th>Time [min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>46</td>
<td>1</td>
<td>69</td>
<td>0.5</td>
<td>0.25</td>
<td>43</td>
<td>4.93</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>383</td>
<td>5</td>
<td>71</td>
<td>3</td>
<td>0.302</td>
<td>40</td>
<td>4.4</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>168</td>
<td>4</td>
<td>75.4</td>
<td>2</td>
<td>0.302</td>
<td>76</td>
<td>2.5</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>174</td>
<td>4</td>
<td>120.6</td>
<td>2</td>
<td>0.416</td>
<td>67</td>
<td>3.47</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>099</td>
<td>2</td>
<td>83.2</td>
<td>1</td>
<td>0.334</td>
<td>43</td>
<td>4.93</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>190</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>47</td>
<td>3.6</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>095</td>
<td>6</td>
<td>83.2</td>
<td>1</td>
<td>0.353</td>
<td>47</td>
<td>23.87</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

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

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 is 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:

\[ X = (x_1, x_2, \ldots, x_N), \]  \hspace{1cm} (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:

\[ \mu_{k,j}(x_k) \in [0, 1], \hspace{0.5cm} (1 \leq j \leq n, \hspace{0.3cm} 1 \leq k \leq N). \]  \hspace{1cm} (2)

The partition matrix is defined as follows:
where the membership values are subject to the following conditions:

\[ \sum_{j=1}^{n} \mu_{kj} = 1 , \ (k = 1 , \ldots , N) . \]  

(4)

The performance index for the partition matrix is given by:

\[ J = \frac{1}{N} \sum_{k=1}^{N} \sum_{j=1}^{n} (\mu_{kj})^w ||x_k - v_j||^2 , \]  

(5)

where \( v_j \) is a pseudo-centre of gravity vector of the \( j \)-th class:

\[ v_j = \frac{\sum_{k=1}^{N} (\mu_{kj})^w x_k}{\sum_{k=1}^{N} (\mu_{kj})^w} , \ (j = 1 , 2 , \ldots , n) \]  

(6)

and \( w > 1 \) is a weight of membership values; \( ||x_k - v_j|| \) is an Euclidean norm used as the distance between \( x_k \) and \( v_j \):

\[ ||x_k - v_j|| = \sqrt{\sum_{i=1}^{n} (x_{ki} - v_{ji})^2} . \]  

(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:

\[ \mu(t) = \begin{bmatrix} \mu_{t1} (x_1) & \cdots & \mu_{tN} (x_N) \\ \vdots & \ddots & \vdots \\ \mu_{tn1} (x_1) & \cdots & \mu_{tnN} (x_N) \end{bmatrix} , \]  

(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:
\[ v'_j = \frac{\sum_{k=1}^{N} (\mu'_{k,j})^w \cdot x_k}{\sum_{k=1}^{N} (\mu'_{k,j})^w}, \quad (j = 1, \ldots, n). \]  (9)

3 – Modify each membership value as follows:

\[ \mu_{k,j}^{t+1}(x_k) = \frac{1}{\sum_{\alpha=1}^{n} \left( \frac{1}{\left\| x_k - v_j^t \right\|^{1/(w-1)}} \right)^{w-1}}, \quad \text{if} \quad x_k \neq v_j^t; \]  (10)

while for \( x_k = v_j^t \), then: \( \mu_{k,j}^{t+1}(x_k) = 1, \quad \text{if} \quad j = \alpha; \quad \mu_{k,j}^{t+1}(x_k) = 0, \quad \text{if} \quad j \neq \alpha. \)

The modified partition matrix is re-organized:

\[
\mu^{t+1}(n) = \begin{bmatrix}
\mu_{11}^{t+1}(x_1) & \cdots & \mu_{1N}^{t+1}(x_N) \\
\vdots & \ddots & \vdots \\
\mu_{n1}^{t+1}(x_1) & \cdots & \mu_{nN}^{t+1}(x_N)
\end{bmatrix}.
\]  (11)

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

\[ \left| \mu_{k,j}^{t+1} - \mu_{k,j}^t \right| < \varepsilon, \]  (12)

where \( \varepsilon \) 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 \( x \), and the cluster centres [the final expressions from (9)], one can calculate the fuzzy degree of the new sample using a relation like (10):

\[ \mu_{k,j}(x) = \frac{1}{\sum_{\alpha=1}^{n} \left( \frac{1}{\left\| x - v_j^t \right\|^{1/(w-1)}} \right)^{w-1}}, \quad \text{if} \quad x \neq v_j^t; \]  (13)

while for \( x = v_j \), then: \( \mu_{j}(x) = 1, \quad \text{if} \quad j = \alpha \) and \( \mu_{j}(x) = 0, \quad \text{if} \quad j \neq \alpha. \)

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 \( p' = x(750 \times 12) \) in chapter 2 (on a column in \( U(7 \times 750) \) 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.

![Graph showing iterative count and objective function values](image)

---

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: \[ U_{100} = \begin{bmatrix} 0.0006 & 0.9990 & 0.0000 & 0.0001 & 0.0002 & 0.0000 & 0.0000 \end{bmatrix}^T \]

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 \( x(750 \times 12) \) namely the most important ones: 4 \( (F_z) \), 6 (Number of intersections), 7 \( (F_z \text{ spectral density}) \) and 10 \( (a_z \text{ spectral density}) \) 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 (c1 is now class 2, and c7 is still 2, which is not correct; c2 is 1, the majority of recordings being on level 1; c3 is divided on levels 1 and 5, which is not acceptable; c4 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.
8. EPUREANU, A., 2005, Simulation, modeling and virtual production methods based on informatics and communication technologies, Contract no. 22 CEEX 103’/05, MEdC.