Tool condition monitoring using artificial intelligence methods

Marek Balazinski,*, Ernest Czogala, Krzysztof Jemielniak, Jacek Leski

Abstract

This paper describes an application of three artificial intelligence (AI) methods to estimate tool wear in lathe turning. The first two are “conventional” AI methods—the feed forward back propagation neural network and the fuzzy decision support system. The third is a new artificial neural network based-fuzzy inference system with moving consequents in if–then rules. Tool wear estimation is based on the measurement of cutting force components. This paper discusses a comparison of usability of these methods in practice.

1. Introduction

Since tool wear has a direct effect on the quality of machined pieces, on-line tool wear monitoring is one of the most important challenges in manufacturing.

Cutting force measurement is one of the most commonly employed methods for on-line tool wear monitoring, especially in turning because cutting force values are more sensitive to tool wear than other measurements such as vibration or acoustic emission (Jemielniak, 1999; Byrne et al., 1995). Cutting force components, however, are also sensitive to the size of the uncut chip cross-sectional area. This area is a product of feed rate and depth of cut, and the value of the latter is not always known. A simple tool-wear estimation method using only information from force components and the feed rate would be very useful in machine shop practice.

Monitoring systems developed in laboratories are often multisensor systems embodying complex AI-based strategies to integrate information, extract features and make more reliable decisions on the state of the tool and process (Balazinski et al., 1994; Byrne et al., 1995; Leski and Czogala, 1997). Recently, artificial neural networks and neuro-fuzzy techniques have been intensively studied and are the most frequently chosen methods of artificial intelligence (AI) for feature fusion (e.g. Balazinski and Jemielniak, 1998; Leski and Czogala, 1997; Jemielniak et al., 1998; Monostori, 1995, 1993). In commercially available systems, however, a “one sensor/one tool per process” approach dominates and only rare applications of AI methods can be found.

The aim of this paper is to compare the use of three AI methods: feed forward back propagation (FF-BP) neural network (FFBPNN), a fuzzy decision support system (FDSS) and an artificial neural network-based fuzzy inference system (ANNBFIS). The focus is not only on the accuracy of the tool-wear prediction but also on practical usability of the presented methods. An important aspect of this usability is the dependence of obtained results on AI system parameters set by the operator.

FF-BP neural network and fuzzy logic systems are rather well known, thus we will give only a short description of the first two methods and a more comprehensive one for the third.

2. Feed forward back propagation neural network

Multi-layer perception (MLP) is one of the best-known types of feed forward neural networks. The MLP with three layers employed in this paper is shown in the upper left corner of Fig. 3. An upper input layer contains three cells, an intermediate hidden layer
contains \( m \) cells and an output layer has one cell. Neurons in the input layer only act as buffers for distributing the input signals (here \( f, F_1 \) and \( F_6 \)) to neurons in the second (hidden) layer. Training of the network is based on a back propagation algorithm using Rumelhart’s generalized delta rule and cumulative weight adjustment. The number of training patterns, \( N_i \) is necessary here and each pattern consists of input values \( f, F_1, F_6 \) and the value of the desired output of network VB. Thus the training set takes the form of a table

\[
f_1 \quad F_{i1} \quad F_{i2} \quad VB_1, \\
f_2 \quad F_{i2} \quad F_{i2} \quad VB_2, \\
\vdots \quad \vdots \quad \vdots \quad \vdots, \\
f_N \quad F_{iN} \quad F_{cN} \quad VB_N.
\]

The patterns are presented consecutively to the network and the desired output values, VB, are compared with the actual ones computed by the network. After presentation of the full set of training patterns in the first iteration, new weight values of the output neuron for the subsequent iteration are calculated.

The artificial neural network system applied in this paper was developed at the Warsaw University of Technology.

3. Fuzzy decision support system

In this section, we present a rule-based approach to decision making using fuzzy logic techniques based on the compositional rule of inference (CRI). This approach is used to handle uncertain or imprecise knowledge and was developed in the 1960s by Zadeh (1973). Such knowledge can be collected and delivered by a human expert (e.g. decision-maker, designer, process planner, machine operator, etc.). The CRI may be written in the form

\[
U' = (C' \times \cdots \times B' \times A') \cdot R,
\]

where \( R \) represents the global relation that aggregates all the rules (knowledge base), \( A', B', \ldots, C' \) represent the inputs (observations) and \( U' \) represents the output (conclusion). The symbol \( \cdot \) represents the CRI operator. Three defuzzification methods are usually available, i.e., center of gravity (COG), average of maximums (AOM) and the modified center of gravity (MCOG). In this paper COG is used for defuzzification. The knowledge base consists of two components; the linguistic term base (database) and the fuzzy production rule base. The database is divided into two parts; fuzzy premises and fuzzy conclusions.

Different combinations of the operations constitute several different variants of the CRI. This is the basis of the inference mechanism used in the FDSS called FUZZY-FLOU. This system was developed at École Polytechnique, de Montréal (Canada), and at the Technical University of Silesia in Gliwice (Poland) (Balazinski et al., 1994).

4. Neural network based-fuzzy inference system

If–then fuzzy rules or fuzzy conditional statements play a pivotal role in approximate reasoning realized using fuzzy systems. Often they are used to capture the human ability to make decisions or control in an uncertain and imprecise environment. Assume that \( m \) numbers of \( n \)-input and one-output (MISO) fuzzy implicative rules or fuzzy conditional statements are given. The \( i \)th rule may be written in the following form:

\[
R^0_i: \text{if } X_1 \text{ is } A^{(i)}_1 \text{ and } \ldots \text{ and } X_n \text{ is } A^{(i)}_n \text{ then } Y = f^0_i(X_1, \ldots, X_n),
\]

where \( X_1, \ldots, X_n \) and \( Y \) are linguistic variables which may be interpreted as inputs of fuzzy system \( (X_1, \ldots, X_n) \) and the output of that system \( Y \). \( A^{(i)}_1, \ldots, A^{(i)}_n \) are linguistic values of the linguistic variables \( X_1, \ldots, X_n \) and \( f^0_i \) is a function of variables \( X_1, \ldots, X_n \).

Taking into account that function \( f^0_i \) is of the form:

\[
f^0_i(x_0) = p^0_i x_0',
\]

where \( x_0' \) denotes an extended input vector \([1 \times 0]^T\). We obtain the final output value in the form (Leski and Czogala, 1997)

\[
y_0 = \frac{\sum_{i=1}^{m} (w^0_i R_i(x_0)/2)p^0_i x_0'}{\sum_{i=1}^{m} (w^0_i R_i(x_0)/2)}
\]

Additionally, we assume that \( A^{(i)}_1, \ldots, A^{(i)}_n \) have Gaussian membership functions:

\[
A^{(i)}_j(x_0) = \exp \left( -\frac{(x_0 - c_j^{(i)})^2}{2s_j^{(i)^2}} \right),
\]

where \( s_j^{(i)}, c_j^{(i)} \); \( j = 1, 2, \ldots, n; i = 1, 2, \ldots, m \) are the parameters of the membership functions.

Eqs. (4) and (5) describe a radial neural network. The unknown parameters (except the number of rules \( m \)) are estimated by means of a gradient method performing the steepest descent on a surface in the parameter space. Therefore a so-called learning set is necessary, i.e. a set of inputs for which the output values are known. This is the set of pairs \((x_0(k), t_0(k)); k = 1, 2, \ldots, N\). The measure of the error of output value may be defined for a single pair from the training set:

\[
E = \frac{1}{2} (t_0 - y_0)^2,
\]

where \( t_0 \) is the desired (target) value of output. The minimization of error \( E \) is made iteratively (for
\( (x)_{\text{new}} = (x)_{\text{old}} - \eta \frac{\partial E}{\partial x} \bigg|_{x=(x)_{\text{old}}} \), \quad (7)

where \( \eta \) is the learning rate.

The system applied in this work was developed at the Technical University of Silesia in Gliwice (Poland). For detailed description of the above neuro-fuzzy system see Leski and Czogala (1997, 1999).

5. Experimental conditions and results

The experiments described in this paper were conducted on a conventional lathe TUD-50. A CSRPR 2525 tool holder equipped with a TiN–Al2O3–TiCN coated sintered carbide insert SNUN 120408 was used in the tests. To simulate factory floor conditions, six sets of cutting parameters were selected and applied in sequence as presented in Fig. 1. Cutting speed in each cut was selected to ensure approximately the same share in tool wear. Tool wear (VB) was measured after carrying out each sequence and its value corresponding to a single cut was linearly interpolated. Cutting forces were measured using a Kistler 9263 dynamometer during 5-s intervals while the cut was executed.

Since the inserts applied in our experiments had a soft, cobalt-enriched layer of substrate under the coating, their tool life had a tendency to end suddenly after this coating wore through. Two experiments were carried out until a tool failure occurred. In the first tests (designated Test W5) 10 cycles were performed until a sudden rise of the flank wear VB occurred, reaching approximately 0.5 mm. In the second test (Test W7) failure of the coating resulted in chipping of the cutting edge at the end of 9th cycle. Flank wear was about 0.35 mm at this point.

Fig. 2 presents the cutting force components; main \( F_c \) and feed \( F_f \), vs. the tool wear obtained in both experiments. Note the weak dependence of the main cutting force \( F_c \) on the tool wear. It is a function of the cutting parameters only

\[ F_c = F_c(a_p, f). \quad (8) \]

This suggests that the cutting force measurement is useless for tool wear estimation. On the other hand, the feed force is independent of feed rate, being affected only by the depth of cut \( a_p \) and the tool wear VB.

\[ F_f = F_f(a_p, VB). \quad (9) \]

Therefore, in order to use feed force measurements to obtain information about tool wear the depth of cut must be identified. This identification can be done directly or, using the weak dependence of cutting force on tool wear

\[ VB = VB(F_c, a_p) \]

or

\[ VB = VB[F_f, a_p(F_c, f)]. \quad (11) \]

This provides an interesting opportunity to estimate the tool wear without requiring information about the depth of cut.

In all three AI methods discussed below, the results of the W5 experiment were used for training or building the knowledge base and the results of the W7 experiment were used for testing. The tool wear assessment was based on values of the feed rate and two cutting force components (cutting force and feed force). Thus, sets of patterns here (4) had the form

<table>
<thead>
<tr>
<th>Test W5 (learning)</th>
<th>Test W7 (testing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_c ) (N)</td>
<td>( F_f ) (N)</td>
</tr>
<tr>
<td>( F_c ) (N)</td>
<td>( F_f ) (N)</td>
</tr>
<tr>
<td>749</td>
<td>273</td>
</tr>
<tr>
<td>577</td>
<td>242</td>
</tr>
<tr>
<td>1254</td>
<td>313</td>
</tr>
<tr>
<td>1120</td>
<td>941</td>
</tr>
<tr>
<td>1177</td>
<td>1018</td>
</tr>
<tr>
<td>2089</td>
<td>1404</td>
</tr>
</tbody>
</table>

For all three AI methods a quality of tool wear estimation was evaluated using average error

\[ \text{rms} = \sqrt{\sum (VB_m - VB_e)^2 / N} \]

and maximum error:

\[ \text{max} = \max (VB_m - VB_e), \]

where \( VB_m \), \( VB_e \) are measured and estimated flank wear, respectively, \( N \) is the number of patterns in set (\( N = 71 \) for W5 and 66 for W7).

6. Application of FF-BP neural network

At first, the artificial neural network was applied for tool wear estimation. The actual structure of this network is presented in Fig. 3.
There are two parameters that have to be set by the operator; the number of neurons (cells) in the hidden layer and the number of iterations. To find out how the number of neurons in the hidden layer influences network performance, training of the networks with 2–10 hidden cells was conducted. In all tests 200,000 iterations were applied. The results obtained are presented in Table 1. For more than 3 cells in a hidden layer the errors are almost the same, which means that the number of these cells can be chosen arbitrarily, say at 5.

The average errors seem to be acceptable for both sets of data. Testing errors are larger of course, and the maximum testing error is extremely large (0.175 mm). This maximum error occurred once, for the last cutting force measurement just after chipping of the cutting edge (see Fig. 2). The chipping did not cause an increase in the flank wear but resulted in an increase in the feed force. Since values for both VB and the chipped edge mean tool failure were high, the discussed answer of the FFBPNN should not be considered as erroneous. Without these last results the maximum error was 0.081 mm and the average was 0.029 mm.

The training duration depends on the capability of the computer used. A 330 MHz Pentium II was used in our case and the training lasted 15.37 min for 5 hidden cells and 28.83 min for 10 hidden cells. Taking into account the fact that the net should be retrained from time to time, this lengthy duration of the training period can be
considered as an important inconvenience of neural network application on the factory floor.

It appeared that courses of errors vs. the number of iterations were almost the same. An example of an FF-BP neural network with 5 hidden neurons is shown in Fig. 3. It can be seen that the weight changes stabilized after some 100,000 iterations. Also, all errors; average (rms) and maximum errors for training and testing sets of data remained almost constant at that point. This means that for the applied number of hidden neurons, the network was not sensitive to overtraining, i.e. it did not fit too closely to the training set of data, which can lead to a loss of generalization ability. In factory floor practice changes of errors should not be shown because this information could be too difficult for the operator to interpret. Despite this evidence that training longer than 100,000 iterations seems to be useless, we decided to leave this iteration number unchanged to ensure that the results were conservative.

7. Application of the fuzzy decision support system

The second AI method applied for tool wear estimation was fuzzy logic. The FDSS described above uses a knowledge base in the form of “if–then” rules to process the information. A set of fuzzy rules expresses a set of fuzzy relations between fuzzy inputs and fuzzy outputs. A certain level of experience and expertise is necessary to develop these rules because some conditions may be uncertain or incomplete and must be estimated. The quality of the rules depends on the quality of the data and therefore on the skills of the expert who generates the data. The standard method used to create a fuzzy knowledge base involves identification of fuzzy inputs and outputs, the elaboration of fuzzy membership functions for each of these inputs and outputs, and finally the construction of fuzzy rules. If the membership functions are constructed properly, usually only a small number of rules are needed.

Fuzzy logic easily models linear and non-linear mathematical functions. This means that fuzzy systems provide interpolative properties. The set of fuzzy rules defines a fuzzy estimation surface. This estimation surface can be visualized as a canopy with a number of peaks where the “ground” represents the fuzzy inputs, the poles represent the rule conclusions (peaks) and the surface of the canopy represents the interpolated fuzzy outputs. The accuracy of the interpolation depends on the number of membership functions, their position, their shape and the rules used to express the relationship between these membership functions. In our paper, a SUM–PROD operator for inference and COA for defuzzification were used because these operators assure a linear interpolation of the output between the rules.

7.1. Construction of fuzzy rules

In our paper, the knowledge base for the decision support system was created directly from the experimental results (W5) presented in Fig. 2. In order to properly compare the results of fuzzy logic with the other AI methods previously described, the same inputs; feed (f), cutting force (Fc) and feed force (Ff), were used as premises, and tool wear (VB) as the conclusion.

From Fig. 2 we can see that the relationship between feed rate (f) and feed force is roughly linear, thus only two fuzzy sets are necessary in this case. Note that in Fig. 4 the extreme values of fuzzy sets for feed are 0.1 and 0.5 mm/rev. These are, respectively, slightly smaller and larger values than used in the W5 experiments (0.17 and 0.47 mm/rev). This widening of the range assures some security margin should the actual feed value input on the shop floor exceed the maximum or minimum values used in experiments.

The same approximately linear relationships can be observed between the cutting force (Fc) and the tool wear (VB), as well as between the feed force (Ff) and the tool wear (VB). As in the previous case only two fuzzy

<table>
<thead>
<tr>
<th>No. of hidden cells</th>
<th>Training time (min)</th>
<th>W5 (training)</th>
<th>W7 (testing)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Average error (mm)</td>
<td>Max error (mm)</td>
</tr>
<tr>
<td>2</td>
<td>7.33</td>
<td>0.0158</td>
<td>0.037</td>
</tr>
<tr>
<td>3</td>
<td>10.43</td>
<td>0.0160</td>
<td>0.037</td>
</tr>
<tr>
<td>4</td>
<td>12.92</td>
<td>0.0140</td>
<td>0.032</td>
</tr>
<tr>
<td>5</td>
<td>15.37</td>
<td>0.0149</td>
<td>0.036</td>
</tr>
<tr>
<td>6</td>
<td>18.03</td>
<td>0.0149</td>
<td>0.035</td>
</tr>
<tr>
<td>7</td>
<td>20.78</td>
<td>0.0150</td>
<td>0.035</td>
</tr>
<tr>
<td>8</td>
<td>23.17</td>
<td>0.0149</td>
<td>0.036</td>
</tr>
<tr>
<td>9</td>
<td>25.72</td>
<td>0.0145</td>
<td>0.034</td>
</tr>
<tr>
<td>10</td>
<td>28.83</td>
<td>0.0144</td>
<td>0.033</td>
</tr>
</tbody>
</table>
sets for both premises are necessary as shown on the right side of Fig. 4.

Following the definition of the premises the fuzzy rules can be written to describe relationships between the premises and conclusions. For three premises with two fuzzy sets each, eight possible rules can be defined. From Fig. 2, Test W5, the first rule can be directly established (see also left side of Fig. 4):

if Feed is 0.1 (mm/rev)
and Cutting Force is 600 (kN)
and Feed Force is 300 (kN)
then Tool Wear VB is 0.1 (mm)

In this rule the fuzziness is expressed by the membership function. The strongest conclusions are at the points of maximum degree of membership (0.1, 600, and 300, respectively). The conclusion value diminishes as the observation gets farther away from the maximum degree of membership.

Some problems occur when we try to define a second rule because experimental results are not available for a feed value of 0.1 mm/rev, cutting force at 600 kN and feed force at 1600 kN. However, taking into account the fact that we only construct a fuzzy estimation surface, we can extend the 1–3 and 5 test curves until they intersect with 1600 kN $F_f$ values. In this case we obtain an approximate value of tool wear equal to 0.9 mm. This value is not used to estimate tool wear. It serves only to complete the integrity of the fuzzy estimation surface.

In the same manner, a set of all rules has been defined. This set of rules is presented in Fig. 4 (left). We can observe that the tool wear is negative in rules 3 and 7. This is due to the extension of the fuzzy estimation surface similar to the process described during construction of rule number two.

After rules definition we can proceed to knowledge base tuning. The objective of tuning is to adjust all knowledge base parameters so that the resulting responses best meet the desired design criteria (e.g. error minimization). It consists of slight adjustments of the shapes and positions of the fuzzy sets. The complexity of the tuning process depends on the number of fuzzy sets and fuzzy rules. In the case of a simple knowledge base (8 rules) as presented in this paper, only minor adjustments were necessary. For example, for the second rule the position of conclusion fuzzy VB set was moved from an initial position of 0.9 to 0.88 mm.

Fig. 4 shows the screen printout of FDSS Fuzzy-Flou with the tuned knowledge base. On the left, a set of rules
in numerical and linguistic form is presented. On the right is a graphical representation of three premises (feed, cutting force and feed force) and one conclusion (tool wear). We can see an example of tool wear estimation. For the following inputs: feed = 0.24 mm/rev, cutting force = 750 kN and feed force = 270 kN, a crisp value (COG of conclusions VB = 0.1 and 0.24 mm) of estimated tool wear is 0.1180 mm.

The final results of the FDSS application to tool wear estimation are presented in Table 2. As in the previous case, a large value of maximum testing error results from chipping of the cutting edge and therefore the answer of FDSS should not be considered as erroneous. Without this last result the maximum error was 0.056 mm and the rms was 0.034 mm.

Unlike a neural network, which is a kind of “black box”, the knowledge base presented above is transparent and understandable. Nevertheless, construction of such a base requires knowledge and experience which can seldom be expected from the machine tool operator. Therefore, fuzzy logic in its “pure” form should not be recommended for small batch production, but rather for mass production where some operations are carried out repeatedly over an extended period of at least several months.

8. Application of an artificial neural network-based fuzzy inference system

The ANNBFIS was applied to the data. As in the case of FFBPNN, there are two parameters that have to be set by the operator. The first is the number of rules, rather than the number of cells as used in the case of FF-BP neural network. The second is the number of iterations; identical to the FF-BP neural network. Several tests were again performed to find out how the number of rules and iterations influence system performance. This time the results were quite different. Firstly, there was almost no dependence on the number of iterations; a similar result was obtained for 2 and 200 iterations. This is the effect of using full optimization (both gradient and least squares) performed in each iteration and preliminary clustering of input data in ANNBFIS system. For the discussed data the ANNBFIS system was fully trained after as little as 2 iterations. Assuming that other cases would likely be more complex, we decided to apply 20 iterations.

The number of rules has some influence on system performance as shown in Fig. 5. It can be seen that optimal results were obtained for 5 rules. Less than 5 rules was not enough to model the learning data, and more was too much, causing a situation where the model fit the data too closely and resulted in a loss of generalization ability.

A very important and convenient feature of the system is a short learning time. Even for the largest number of rules (10) and iterations applied (200), training lasted for < 1 min. For 3 rules and 20 iterations, the learning time is just 3 s. This makes the system easy to use and retraining can be done as required at any time, which enables the creation of an effective optimizing program.

9. Comparison of applied AI methods

The measured tool wear (VB) vs. time in both tests is compared with values estimated using all three discussed AI methods in Fig. 6. Table 2 presents the errors of these estimations; note that the last measurement in test W7 was not included because its high value resulted from a catastrophic tool failure. The results obtained with all
systems are acceptable and all three systems would fire end of tool life alarms at the same time. The differences in estimated tool wear from each method are so small that one cannot readily draw a conclusion regarding which is better for tool conditions monitoring. Much more important are the differences in practical convenience of their application on the factory floor, as discussed in previous sections.

10. Conclusions

All three AI methods applied in this work to estimate tool wear gave similar, acceptable results. Important differences in the “internal” structure of these systems would be unknown and thus irrelevant for the operator. A major difference in their usage, however, is a critical factor.

Construction of a knowledge base for the fuzzy logic system necessitates skill and expert knowledge. The operator has to analyze the dependence of cutting forces on tool wear, which means that the results of preliminary experiments have to be presented to the operator in a convenient and transparent form. This makes fuzzy logic rather difficult for practical implementation.

The number of neurons in the hidden layer of the neural network and the number of iterations can be selected arbitrarily as they have very little influence on system performance. The results of preliminary (learning) tests do not have to be presented to the operator in this case. Instead, they are simply fed to the system input. The disadvantage of the method is a considerably long training time, which makes it inconvenient in practice.

Similarly for the neuro-fuzzy system, the structure (number of rules) and the number of iterations do not have an important influence on system performance and the operator does not have to know the results of preliminary tests. The most important difference between these last two methods is learning time; it is so short for the neuro-fuzzy system that it can be easily optimized and implemented on the factory floor.

References


