Tool wear monitoring using genetically-generated fuzzy knowledge bases

Sofiane Achichea, Marek Balazinski,a, Luc Baron*a,*, Krzysztof Jemielniakb

a Department of Mechanical Engineering, École Polytechnique de Montréal C.P. 6079, succ. CV, Montréal, Québec, Canada H3C 3A7
b Faculty of Production Engineering, Warsaw University of Technology, Narbutta 86, 02-524 Warsaw, Poland

Abstract

Fuzzy logic is an AI method that is being implemented in a growing number of different fields. One of these applications is tool wear monitoring. The construction of a fuzzy knowledge base from a set of experimental data by a human expert however, is a time consuming task, and hence, limits the expansion of the use of this AI method. Alternatively, the fuzzy knowledge base can be automatically constructed by a genetic algorithm from the same set of experimental data without requiring any human expert. This paper compares these two fuzzy knowledge base construction methods and the results obtained in a tool wear monitoring application.

Keywords: Artificial intelligence; Fuzzy decision support system; Knowledge base; Tool condition monitoring; Genetic algorithm

1. Introduction

Since tool wear has a direct effect on the quality of machined parts, on-line wear monitoring is one of the most important challenges in manufacturing. Tool wear influences a variety of machining phenomena and thus most monitoring systems use an easily measured parameter, such as the increase of the cutting force or other related quantities, as a basis for tool wear estimation (Byrne et al., 1995; Du et al., 1995; Jemielniak and Kosmol, 1995). Systems developed in laboratories are often multi-sensor groups embodying artificial intelligence (AI) methods in order to make a more reliable estimation of the state of the tool and consequently of the machined parts themselves (Byrne et al., 1995; Balazinski et al., 1994; Leski and Czogala, 1997). Usually a set of experimental tests involving different cutting conditions, e.g., different feed rate and depth of cut, is repeatedly performed on a typical part. During machining, the cutting and feed forces are recorded while the tool wear is manually measured after each test. This experimental data is then cast into a knowledge base (KB) through a learning process. Finally, this KB is used by an AI method to predict tool wear. Among these methods, fuzzy logic (FL) systems, neural networks (NN) and neuro-fuzzy (NF) systems are the most frequently chosen, for this type of application (Leski and Czogala, 1997; Jemielniak, 1999; Monostori, 1993; Monostori, 1995). The aim of this paper is to compare performances of two FL-based monitoring systems relative to those of a NN-based system for application to tool wear estimation. The first FL system, called FL-MA, uses a KB manually constructed by a human expert from a set of experimental data, while the second FL system, called FL-GA, uses a KB automatically constructed by a genetic algorithm (GA) from the same set of experimental data. Many research works have been conducted concerning the automatic generation of fuzzy knowledge bases. Some of these have focused on different aspects of the automatic generation of fuzzy rules using GAs (Thrift, 1991; Diederich and Renaud, 1999; Hagras et al., 1999; Yuan and Zhuang, 1995; Valenzuela-Rendon, 1991; Janikow, 1995). Conversely, other works have focused on automatic generation of fuzzy sets with GAs (Valesco et al., 1997; Nomura et al. 1992). While very few have focused on simultaneous generation, a very simple GA was used for this purpose in Homaiifar and
McCormick (1995). The GA used in this paper has been customized to achieve a KB that is accurate, yet simple in structure (small rule set and fuzzy set distribution). It has been developed by the authors at Ecole Polytechnique of Montreal (Baron et al., 2001). The GA tunes all the components of the KB, i.e. distribution and number of fuzzy sets, the fuzzy-rule base (the rules themselves) and the number of rules. The FL system, called Fuzzy-Flou, has been developed by the authors in collaboration with the Warsaw University of Technology. The experimental conditions of tool wear estimation are described together with the learning and operating conditions. Finally, performances and required resources are compared and discussed.

2. Monitoring systems

Since tool wear monitoring requires multiple input information to predict the tool wear, this type of system can be cast into the class of multi-input and single output (MISO) systems.

2.1. Fuzzy logic system

A rule-based approach to decision making using FL techniques may consider imprecise vague language as a set of rules linking a finite number of conclusions. The knowledge base of such systems consists of two components: a linguistic terms base and a fuzzy rules base (Balazinski et al., 1994). The former is divided into two parts: the fuzzy premises (or inputs) and the fuzzy conclusions (or outputs). For the sake of simplicity, we consider only non-symmetric triangular fuzzy sets on \( n \) inputs (called premises in this context) and sharp-symmetric triangular fuzzy sets on the single conclusion. The representation of such imprecise knowledge by means of fuzzy linguistic terms makes it possible to carry out quantitative processing in the course of inference that is used for handling uncertain (imprecise) knowledge. This is often called approximate reasoning (Zadeh, 1973). Such knowledge can be collected and delivered by a human expert (e.g. decision-maker, designer, process planner, machine operator). This knowledge, expressed by \((k = 1, 2, \ldots, K)\) a finite number of heuristic fuzzy rules of the type MISO, may be written in the form:

\[
R^{\text{MISO}}_k: \quad \text{if } x_1 = X_1^k \text{ and } x_2 = X_2^k \text{ and } \ldots \text{ and } x_n = X_n^k \text{ then } y = Y^k, \tag{1}
\]

where \{\(X_i^k\)\}_{i=1}^n denote values of linguistic variables \{\(x_i\)\}_{i=1}^n (conditions) defined in the following universe of discourse \{\(X_i\)\}_{i=1}^n; and \(Y^k\) stands for the value of the independent linguistic variable \(y\) (conclusion) in the universe of discourse \(Y\). The global relation aggregating all rules from \(k = 1\) to \(K\) is given as

\[
R = \bigcup_{k=1}^K (R^{\text{MISO}}_k), \tag{2}
\]

where the sentence connective \(\text{also}\) denotes any \(\lor\) or \(\land\) norm (e.g., \(\min(\land)\) or \(\max(\lor)\) operators) or averages. For a given set of fuzzy inputs \{\(X_i^k\)\} (or observations), the fuzzy output \(Y\) (or conclusion) may be expressed symbolically as:

\[
Y = (X'_1, X'_2, \ldots, X'_n) \circ R, \tag{3}
\]

where \(\circ\) denotes a compositional rule of inference (CRI), e.g., the \(\text{sup}\land\) or \(\text{sup}\lor\) (also denoted \(\text{sup}\ast\)). Alternatively, the CRI of Eq. (3) is easily computed as

\[
Y = X'_n \circ \ldots \circ (X'_2 \circ (X'_1 \circ R)). \tag{4}
\]

Usually, there are four variants of CRI: the sentence connective \(\text{also}\) can be either \(\lor\) or \(\land\) norm (\(\sum\)); the compositional operator is the supremum (sup) of either \(\land\) or \(\ast\), denoted \(\text{sup}\land\) and \(\text{sup}\ast\); while the sentence connective \(\land\) and the fuzzy relation are always identical to the second part of the latter. For the sake of brevity, all four variants of CRI—i.e.: \(\lor\text{-sup}\land\text{-}\ast\ast\ast\ast\ast\ast\); \(\land\text{-sup}\ast\ast\ast\ast\ast\ast\); \(\text{sum}\text{-}\ast\ast\ast\ast\ast\ast\); and \(\text{sum}\text{-}\ast\ast\ast\ast\ast\ast\)—are expressed as

\[
Y = \left\{ \begin{array}{l}
\bigvee_{k=1}^K \text{sup}_{\{x_i \in X_i^k\}} \ast_{\{X'_i, \ldots, X'_i, X'_i\}} \ast_{\{X_1^k, X_2^k, \ldots, X_n^k, Y^k\}} \end{array} \right. \tag{5}
\]

where \(\ast_{\{\cdot\}}\) denotes the t-norm of \(\{\cdot\}\) defined as either \(\land\) or \(\ast\). These variants of CRI mechanisms allow us to obtain different conclusions represented as the membership function \(Y\). Additionally, there are usually three defuzzification methods: the centre of gravity (COG); the mean of maxima (MOM); and the height method (HM). All the results presented in this paper are obtained with the \(\text{sum}\text{-}\ast\ast\ast\ast\ast\ast\) CRI and COG as defuzzification. As shown in Fig. 1, the fuzzy knowledge base, which is the heart of such a system, must be constructed either manually by a human expert, as described in Section 3.1, or automatically by a learning process like GA, as described in Section 3.1.

2.2. Genetic algorithm

GAs are powerful stochastic optimization techniques that are based on the analogy of the mechanics of biological genetics and imitate the Darwinian survival-of-the-fittest approach (Baron, 1998). As shown in Fig 2, each individual of a population is a potential FDSS Fuzzy-Flou KB. The method uses iterative improvement of individuals at each generation to converge toward multiple optima simultaneously. This evolutionary process operates directly on the genotype—i.e., the coded physical characteristics into bit string—of individuals rather than on its phenotype—i.e., the physical characteristics themselves. It is noteworthy that
the coding of several parameters into bit strings is crucial in GA. When the number of unknown parameters increases, GA exhibits only a polynomial increase in the size of the search space, while the other optimization techniques show an exponential increase. Fig. 2 presents the encoding/decoding scheme as well as the four basic operations, i.e.: reproduction, mutation, evaluation and natural selection, of the developed GA learning software (Achiche et al., 2000).

2.2.1. Coding

The genotype of a fuzzy KB is the coding of the fuzzy sets and rules into a bit-string:

\[ G = \{G_{\text{sets}}, G_{\text{rules}}\}. \] (6)

where \(G_{\text{sets}}\) and \(G_{\text{rules}}\) are respectively the genotypes of the fuzzy sets and rules.

2.2.1.1. Fuzzy set coding. The FDSS Fuzzy-Flou allows the use of trapezoidal membership functions, as shown in Fig. 3. For the sake of coding simplicity, we consider only non-symmetrical triangular fuzzy sets for premises and symmetrical triangular fuzzy sets for the conclusion. Therefore the position of each fuzzy set is given as \(m_1 = m_2 = \) position of the summit, \(a_m\) and \(b_m\) are set to reach the positions of the previous and next summit,
while \( h_m = 1 \) (see Fig. 3). The size of these \( G_{\text{sets}} \) depends on the number of premises \( N \), the number of fuzzy sets \( N_i \) on each premise \( i \) and the number of bits \( b_r \) allocated to specify the resolution on the position. For example, if \( b_r = 4 \), the genotype of the fuzzy sets of premise \( i \) is given as

\[
G_{Xi} = \left\{ \begin{array}{c} 1001 \\ \text{summit}_1 \\ 1110 \\ \text{summit}_2 \\ \vdots \\ \text{summit}_{K_i} \end{array} \right\},
\]

where \( K_i \) is the number of fuzzy sets on premise \( Xi \) excluding the two summits located at the extreme values of each premise. The total size of \( G_{\text{sets}} \) is given as

\[
\text{size}(G_{\text{sets}}) = \sum_{i=1}^{N} K_i b_r + K_y b_r,
\]

where \( K_y \) is the number of fuzzy sets on the conclusion. However on the conclusion the number of fuzzy sets is equal to the number of the coded summits since the limits are also coded.

2.2.1.2. Fuzzy rules coding. The genotype of fuzzy rules must contain information about all the possible combinations connecting one fuzzy set on each premise to a fuzzy set on the conclusion. For \( N \) input premises and \( K_i \) fuzzy sets on premises, the maximum number of fuzzy rules \( K \) is computed as

\[
K = (K_1 + 2) \times (K_2 + 2) \times \cdots \times (K_N + 2).
\]

As shown in Fig. 4, the fuzzy rules are coded as an ordered list of combination of the premises, each having an enable/disable bit, denoted \( e \)—0 for disable; 1 for enable—together with a conclusion fuzzy set number. Each rule is coded into a 4 bit string, i.e.,

\[
G_{\text{rules}} = \left\{ e_{\text{rule}_1} e_{\text{rule}_2} \cdots e_{\text{rule}_K} \right\},
\]

2.2.2. Reproduction mechanisms

The evolution of the population is achieved by reproduction of the best individuals based on their ability to survive natural selection. This reproduction is performed with a combination of the four following operators.

2.2.2.1. Simple crossover. The reproduction is mainly performed by crossing of the genotype of the parents, in order to obtain the genotype of two children. One of the techniques of crossover is shown in Fig. 5. This part of the mechanism is governed by an initiating probability \( p_1 \).

2.2.2.2. Fuzzy sets displacement. Displacement of the fuzzy sets is performed (with a probability \( p_2 \)) by randomly selecting a fuzzy set on a premise. The selected fuzzy set is then moved by one step of resolution toward the left or right, with an equal probability (see Fig. 6). This reproduction operator has the virtue of trying different fuzzy set repartitions, while decreasing the number of fuzzy sets by superimposing two or more of them.

2.2.2.3. Fuzzy-rules reduction. The reduction of the number of fuzzy rules is performed with a probability \( p_3 \) given by

\[
p_3 = 1 - p_1 - p_2.
\]

One of the \( K \) fuzzy rules is randomly selected and deactivated—the bit \( e \) is set to disable—as shown in Fig. 7. Obviously, this reproduction operator does not always generate a reduction in the number of fuzzy rules, but gradually works in that direction. The bias toward the reduction of the number of rules tends to produce small KBs.

2.2.2.4. Mutation. Mutation is a random inversion of a bit in the genotype of a new member of the population as shown in Fig. 8. Mutation makes it possible to try
completely different solutions. The probability of mutation $p_4$ should be kept very small in order to give the other reproduction operators precedence for improving the population.

This way of seeking completely different solutions allows the algorithm to jump out of a local optimum, and potentially fall into more promising regions.

2.2.3. Natural selection and evaluation

The capacity of each KB to survive natural selection is measured by two objective functions. The first objective function, denoted $\phi_1$, evaluates the capacity of a KB to approximate the set of experimental data, i.e.

$$\phi_1 = \frac{\delta - \varepsilon_{\text{RMS}}}{\delta}, \quad \delta = p_Y^{\text{max}} - p_Y^{\text{min}},$$

(12)

where $\delta$ is defined as the range on the conclusion $Y$ and $\varepsilon_{\text{RMS}}$ the root-mean-square error between the fuzzy
conclusion \( Y_i \) computed with the FKB and the corresponding experimental conclusion \( y_i \), for the \( N \) experimental data, i.e.

\[
\varepsilon_{\text{RMS}} = \sqrt{\frac{\sum_{i=1}^{N} (Y_i - y_i)^2}{N}}.
\]

The second objective function, denoted \( \phi_2 \), evaluates the complexity of a KB through its number of active fuzzy rules, i.e.,

\[
\phi_2 = \frac{K - n_a}{K},
\]

where \( K \) is recalled to be the maximum number of fuzzy rules and \( n_a \) the actual number of active fuzzy rules. In order to deal with these two contradictory objectives, a weighted sum of the two preceding objective functions is used, i.e.,

\[
\phi = w\phi_1 + (1 - w)\phi_2,
\]

where the weight \( w \) is usually set around 75%. The influence of \( o_1, p_1, p_2 \) and \( p_3 \) are extensively discussed in Balazinski et al. (2000). Natural selection is performed on the population by keeping the most promising KB along a single fitness value. This is equivalent to using solutions that are closest to the optimum. In this work, the size of the population is kept constant at 100 KB (100 being the usual population size). At each generation, the reproduction produces 100 brand new KB and the 200 resulting KB (100 + 100 = 200) are evaluated and ranked. The natural selection applies on the 200 KB by keeping the 50 best non-identical KB along with \( \phi \) and \( \phi_1 \), for a total of 100 KB (50 + 50 = 100). The GA learning process is terminated after a specific time period; hence for each learning a pretest is run to set up the maximum number of generations that will influence both learning time and accuracy.

3. Knowledge base learning and results

In this paper, tool wear, denoted \( VB \), is estimated from \( n = 3 \) input information, i.e.: the feed rate, denoted \( f \); the feed force, denoted \( F_f \); and the cutting force, denoted \( F_c \). This choice of input variables is based on the following two observations (see Fig. 9): Force \( F_t \) is independent of \( f \), but rather depends on \( VB \) and the depth of cut, denoted \( d \). Moreover, \( F_c \) depends on \( d \) and \( f \), while being only weakly dependent on \( VB \). This relationships provides interesting opportunities to use \( f \) and the measurement of \( F_c \) to determine \( d \), and the use of the measurement \( F_t \) to estimate \( VB \) without requiring \( d \) as input variable. Two sets of experimental tests were repeatedly performed until a tool failure occurred. Test W5 was devised for KB learning, while test W7 was used to verify the performances of the different monitoring systems. In order to simulate factory floor conditions, a typical part was machined on a conventional lathe under six different cutting conditions as shown in Fig. 10. The cutting speed of each operation was selected to ensure approximately the same rate of tool wear. Tool wear was manually measured after carrying out each cycle and the \( VB \) of each single cut was linearly interpolated. For each cut, \( f \) and \( F_c \) were measured using a Kistler 9263 dynanometer during 5 s intervals while the cut was executed. Since the inserts used in the experiments had a soft, cobalt-enriched layer of substrate under the coating, the tool life had a tendency to end suddenly after this coating wore through. In test W5, ten cycles were performed until a sudden rise of flank wear \( VB \).
occurred, reaching approximately 0.5 mm. In test W7, failure of the coating resulted in chipping of the cutting edge at the end of the 9th cycle; flank wear at this point was about 0.35 mm. The results of tests W5 and W7 are shown in Fig. 9. The approximation error of the monitoring systems is measured using the root-mean-square error

\[ A_{\text{rms}} = \sqrt{\frac{\sum_{i=1}^{N} (VB_m - VB_e)^2}{N}}, \]  

and the maximum error

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

where \( VB_m \) and \( VB_e \) are, respectively, the measured and estimated \( VB \) and \( N \) is the number of patterns in the experimental test (i.e., \( N = 71 \) for W5; and \( N = 66 \) for W7).

For the sake of comparison, the first FKB has been manually constructed by a human expert with the experimental data as described below. The second FKB has been automatically constructed using a GA with the same set of experimental. Finally, the results obtained with these two FKBs are compared with those of a neural network, as reported in Balazinski et al. (2002).

### 3.1. Manually-constructed fuzzy knowledge base

A certain level of experience and expertise is required in order to manually develop a fuzzy knowledge base (FKB) from a set of experimental data since some conditions may be uncertain and incomplete, and hence must be estimated. Moreover, there is no systematic way to transform the knowledge of a human expert into a knowledge base of a FDSS system. The quality of the FKB depends on the quality of the data and the skills of the expert. The usual approach is to choose the number and location of the fuzzy membership functions on each premises and the conclusion, and then determine the fuzzy rules. If the membership functions are wisely chosen only a small number of fuzzy rules are usually needed.

The relationship between \( F_t \) and \( VB \), as shown in Fig. 9, is roughly linear, and thus only two fuzzy sets are necessary on this premise. The same approximately linear relationship can be observed between \( F_s \) and \( VB \) for each specific cut depth; 1.5 and 3 mm. Hence, only two fuzzy sets are necessary for these two premises as shown on the right of Fig. 11. It is noteworthy that the extreme values of the two fuzzy sets on premise \( f \) are 0.1 and 0.5 mm/rev; providing a wider range than was used in the experiments (i.e., 0.17 and 0.47 mm/rev). This widening of the feed rate range ensures that the monitoring system works properly in cases when shop floor feed rate values exceed the experimental range.

There are a maximum of eight possible fuzzy rules with two fuzzy sets on three premises (i.e. \( 2^3 = 8 \)). From Fig. 9, test W5, the first fuzzy rule can be directly established as: If \( f \) is 0.1 mm/rev and \( F_s \) is 600 N and \( F_t \) is 300 N then \( VB \) is 0.1 mm. This is shown on the left side of Fig. 11. In such a rule, the fuzziness is expressed by the membership function. The strongest conclusions arise at the maximum degree of memberships of each premise (i.e., 0.1, 600 and 300, respectively). The conclusion value diminishes as the observations move away from their maximum degree of memberships. Problems occurred when the expert tried to define a second fuzzy rule because there was no experimental measure of \( VB \) for \( f \) = 0.1 mm/rev, \( F_s \) = 600 N and \( F_t \) = 1600 N. However it was possible to extrapolate the actual measurements up to an \( F_s \) = 600 N. In this case we obtained an approximate value of \( VB = 0.9 \) mm. Obviously, this value of \( VB \) does not have any physical meaning. It only serves to complete the integrity of the FKB. As shown for the fuzzy rules 3 and 7 of Fig. 11, the \( VB \) can even be negative for this purpose. Once all the fuzzy rules are defined, the expert can perform a tuning of the location of the fuzzy sets on the premises together with a revision of the fuzzy rules themselves with the aim of reducing the approximation errors. The complexity of this tuning process depends on the number of fuzzy sets and rules. For a simple FKB only minor adjustments should be necessary. In our case here, the location of the conclusion fuzzy set of the fuzzy rule 2 was moved from \( VB = 0.9 \) to 0.88 mm.

Fig. 11 shows a screen printout of the software Fuzzy-Flou developed at École Polytechnique de Montréal and the Technical University of Silesia in Gliwice with the manually-tuned FKB. On the left side, the fuzzy rules in numerical and linguistic forms are presented. On the right side, the fuzzy sets are shown on the three premises. One can see an example of \( VB \) estimation for the following inputs: \( f = 0.24 \) mm/rev, \( F_s = 750 \) N and \( F_t = 270 \) N. A crisp value (centre of gravity of two weighted conclusions \( VB = 0.1 \) and 0.24 mm) of estimated \( VB \) is 0.118 mm.

The performance results of the FL-MA system are presented in Table 1. A large value of \( A_{\text{max}} \) results from
chipping of the cutting edge, and therefore the answer of the FL-MA should not be considered erroneous. Without this last result $\Delta_{\text{max}}$ is 0.056 mm and $\Delta_{\text{rms}}$ is 0.034 mm. Unlike the NN, which is a kind of black-box, the FKB presented above is transparent and understandable. Nevertheless, the manual construction of the FKB requires knowledge and experience, which cannot always be expected from a machine tool operator. Therefore, FL in its pure form is not recommended for small batch production, but rather for mass production where some operations are carried out repeatedly over an extended period of time; at least several months.

3.2. Genetically-constructed fuzzy knowledge base

A genetic algorithm can be used to automatically generate the FKB from the same set of experimental data (test W5). The GA uses a set of probabilities (i.e., $p_1, p_2, p_3, p_4$) in order to control the occurrences of the different reproduction operators. Obviously, a different set of values drive the GA toward an FKB with different behaviours. Moreover, changing the weight $\omega$ between the two contradictory objectives $\phi_1$ (approximation error) and $\phi_2$ (number of fuzzy rules) produces FKBs with completely different behaviours.

The value of weight (called $\omega$) used at each iteration of the learning process is generally different from the one used at the end of the learning process to select the final FKB (called $\omega_f$). Four FKB have been automatically constructed from test W5 with the following parameters:

- Run 1: $\omega = 0.80$, $p_1 = 0.85$, $p_2 = 0.13$.
- Run 2: $\omega = 0.80$, $p_1 = 1.00$, $p_2 = 0.00$.
- Run 3: $\omega = 1.00$, $p_1 = 0.85$, $p_2 = 0.13$.
- Run 4: $\omega = 1.00$, $p_1 = 1.00$, $p_2 = 0.00$.

It is noteworthy that $\omega = 1.00$ puts all the emphasis on the approximation accuracy, while $\omega = 0.00$ puts the

<table>
<thead>
<tr>
<th>Table 1</th>
<th>$\Delta_{\text{rms}}$ of the three AI methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI method</td>
<td>$\Delta_{\text{rms}}$ (mm)</td>
</tr>
<tr>
<td>Neural network</td>
<td>0.015</td>
</tr>
<tr>
<td>FL-MA</td>
<td>0.024</td>
</tr>
<tr>
<td>FL-GA</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Fig. 11. Screen printout of the manually constructed knowledge base.
emphasis on a reduction in the number of fuzzy rules.
The other parameters are fixed as follow:

- \( p_4 = 0.05 \)
- \( o_3 = 1.00 \)

The operator must define the maximal limits of complexity of the desired FKB. These limits are chosen for this problem as:

- a maximum of 7 fuzzy sets on each premise;
- a maximum of 8 fuzzy sets on the conclusion.

As a result, the maximum number of fuzzy rules is given as: \( 7 \times 7 \times 7 = 343 \). As shown in Table 2, \( D_{\text{rms}} \) and \( D_{\text{max}} \) are both acceptable; note that several runs (learning) with identical evolution parameters converge to the same KB, similar to the results obtained by the neural network for the two experiments (W5 and W7). The \( D_{\text{max}} \) is slightly high because of the last measurement of \( VB \), as was also noticed with the other systems. The time of the execution is around 14 min for each run (on Pentium II-350 Mhz), which makes the method very attractive for factory floor use. The best approximation accuracies are obtained in Run 4; both \( D_{\text{rms}} \) and \( D_{\text{max}} \) are relatively low and in the same range as those obtained by the neural network method. However, the FKB requires 38 fuzzy rules. Conversely, Run 2 provides an FKB with only 4 fuzzy rules with of course a higher approximation error. It is noteworthy that as many as 38 fuzzy rules are still manageable by a human expert. Recall that this number of fuzzy rules has been reduced by the GA from the maximum of 343 fuzzy rules.

4. Comparison and remarks

All three AI methods used to estimate tool wear give satisfactory results. There is a slight difference in the

<table>
<thead>
<tr>
<th>Number of fuzzy rules</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>W5 (training)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( D_{\text{rms}} ) (mm)</td>
<td>0.041</td>
<td>0.040</td>
<td>0.050</td>
<td>0.020</td>
</tr>
<tr>
<td>( D_{\text{max}} ) (mm)</td>
<td>0.110</td>
<td>0.110</td>
<td>0.170</td>
<td>0.070</td>
</tr>
<tr>
<td>W7 (testing)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( D_{\text{rms}} ) (mm)</td>
<td>0.040</td>
<td>0.050</td>
<td>0.064</td>
<td>0.037</td>
</tr>
<tr>
<td>( D_{\text{max}} ) (mm)</td>
<td>0.090</td>
<td>0.140</td>
<td>0.170</td>
<td>0.100</td>
</tr>
</tbody>
</table>

Fig. 12. Knowledge base obtained for Run 1.
approximation errors depending on the set of control data used (W5 and W7). The testing set (W5) is approximated with a lower level of error, which was predictable since it is the set used for training.

An interesting observation can be made from the automatically constructed FKB (see Table 2). Even if Run 4 gives the best approximation error, since $\Delta_{\text{rms}}$ is lowest for W7, the difference between the $\Delta_{\text{rms}}$ of W5 and W7 increases with the complexity of the FKB. Run 1 and 2 are the simplest, providing less fuzzy rules than runs 3 and 4, which are more complex (see Table 2). This can be explained by a too close approximation of the training set (essentially the case in Run 4) that leads to the generation of a specific FKB, rather than a model that can be used for other sets of data.

Figs. 12 and 13 show respectively the knowledge bases corresponding to Run 1 and 4. If a fuzzy knowledge base has to be selected from the four runs, Run 4 would be an obvious choice since it provides the lowest $\Delta_{\text{rms}}$ (an error near to those provided by the two other methods—FL-MA and NN—see Table 1). However, it is more interesting to use the results of Run 1 instead of the others since it offers a better balance between simplicity and accuracy of the FKB. We can also notice its stability, as the $\Delta_{\text{rms}}$ stays at the same value for W5 and W7. Run 2, which gives the simplest FKB, is not reliable, since it uses a small number of fuzzy rules, leaving an important part of the input domain uncovered by fuzzy rules (lack of information), even thought it provides good results for both sets of data (W5 and W7). Such a small amount of fuzzy rules remains a risky choice.
Acknowledgements

Financial support from the Natural Sciences and Engineering Research Council of Canada under grants of RGPIN-203618 and RGPIN-105518 is gratefully acknowledged.

References


