

C. Scheffer · H. Engelbrecht · P. S. Heyns

## A comparative evaluation of neural networks and hidden Markov models for monitoring turning tool wear

Received: 5 November 2004 / Accepted: 4 March 2005 / Published online: 14 June 2005  
© Springer-Verlag London Limited 2005

**Abstract** Condition monitoring of machine tool inserts is important for increasing the reliability and quality of machining operations. Various methods have been proposed for effective tool condition monitoring (TCM), and currently it is generally accepted that the indirect sensor-based approach is the best practical solution to reliable TCM. Furthermore, in recent years, neural networks (NNs) have been shown to model successfully, the complex relationships between input feature sets of sensor signals and tool wear data. NNs have several properties that make them ideal for effectively handling noisy and even incomplete data sets. There are several NN paradigms which can be combined to model static and dynamic systems. Another powerful method of modeling noisy dynamic systems is by using hidden Markov models (HMMs), which are commonly employed in modern speech-recognition systems. The use of HMMs for TCM was recently proposed in the literature. Though the results of these studies were quite promising, no comparative results of competing methods such as NNs are currently available. This paper is aimed at presenting a comparative evaluation of the performance of NNs and HMMs for a TCM application. The methods are employed on exactly the same data sets obtained from an industrial turning operation.

The advantages and disadvantages of both methods are described, which will assist the condition-monitoring community to choose a modeling method for other applications.

**Keywords** Neural networks · Hidden Markov models · Condition monitoring · Tool wear

### 1 Introduction

Monitoring the condition of machine tools is important to ensure quality, reliability and availability in production environments. Machine tools (e.g. metal removal machines such as CNC lathes and milling machines) are common in mass production environments and have to be utilized fully to justify the investment of capital and running costs. Such machines and their components can be monitored in various ways to prevent failures, for instance, by monitoring the gearboxes, bearings, hydraulic fluids and other components. However, it would be extremely useful if the cutting tools themselves could be directly monitored. One factor that continues to thwart the perfect automation of machine tools in mass production environments is that all cutting tools are prone to wear.

The shape (mode) that the wear produces on tool inserts and also the rate of wear are part of a complex process and it is generally accepted that analytical models have limited accuracy in such applications. Recent advances in numerical methods are promising but the chaotic way in which tool wear forms still limits these methods to new (and consequently perfectly sharp) tools only. Numerical methods are typically nonlinear finite element models (FEM) that can yield the cutting forces and temperatures during a metal cutting operation.

The remaining options are online monitoring (i.e. indirect estimation) and direct measurement of the tool wear. A direct measurement of tool wear is

---

C. Scheffer (✉)  
Design and Mechatronics  
Division Department of Mechanical Engineering,  
University of Stellenbosch,  
Stellenbosch, 7602, South Africa  
E-mail: cscheffer@sun.ac.za  
Tel.: +27-21-8084249  
Fax: +27-21-8084958

H. Engelbrecht  
DSP Research Group Department of Electronic Engineering,  
University of Stellenbosch, Stellenbosch,  
7602, South Africa

P. S. Heyns  
Dynamic Systems Group Department of Mechanical  
and Aeronautical Engineering,  
University of Pretoria, Pretoria, 0002, South Africa

time-consuming and historical developments to achieve this have achieved only partial success. The sole practical option left is to make use of sensors to achieve an indirect estimation of the tool wear. Typical sensor approaches are the use of power, force, vibration, and acoustic emission (AE). Sensory information is processed and correlated with the known values of tool wear. Models of tool wear can be established by using empirical data in this way. Various researchers have attempted this approach. In fact, the literature on the subject of TCM amounts to hundreds of research papers over the past decade, describing numerous methods applied to processes such as turning, milling, drilling, and grinding [1, 2]. Due to the complex relationship between signal features and the dynamics of the machining process, it is also commonly accepted that techniques such as NNs should be used to model the relationship between sensory information and tool wear. Neural Networks are generally used because they can model complex input–output relationships even when using noisy and incomplete data to train them.

Some papers have recently shown that Hidden Markov models (HMMs) are also extremely useful for the purpose of modeling sensory information in condition monitoring and tool wear applications [3, 4, 5]. The training algorithm used for the parameter estimation of HMMs is called the Baum-Welch algorithm, which is an Expectation-maximization algorithm. The advantage of using the Baum-Welch algorithm is that it is guaranteed to converge to a local optimum. Furthermore, given a set of training data and an initialized HMM, the Baum-Welch algorithm will always result in exactly the same estimated HMM parameters. Therefore, the quality of the HMM modeling is determined by the amount of training data available and the initialization of the HMMs. If the initialization is done in a deterministic manner (and not randomly) one will always obtain the exact same model after training. Hence the question remains, are HMMs or NNs preferable for online and indirect estimation of tool wear during a machining operation? This paper describes how data from an industrial turning operation was processed with both techniques in order to compare the results directly. Some background is given on both techniques, with reference to recent applications in the field of tool wear monitoring.

### 1.1 Neural networks

Neural network modeling is ideal for TCM problems because it utilizes a matrix of independent data simultaneously to make a classification. Two of the most attractive characteristics of NNs are their extraction of underlying information and robustness regarding distorted sensor signals. This also applies to sensor fusion schemes for TCM. Combining the features from the vibration, AE, force, and current signals, results in a model that can more accurately predict the tool

condition [6]. The successful implementation of NNs depends on properly selecting the network structure as well as on the availability of reliable training data. It is also important to distinguish between supervised and unsupervised network paradigms. Unsupervised NNs are trained with input data only and are normally used for the discrete classification of different stages of tool wear. Supervised NNs are trained with input and output data and are used for a continuous estimation of tool wear.

There are two basic network paradigms for unsupervised classifications, namely Adaptive Resonance Theory (ART) and the Self-Organizing Map (SOM), also known as the Kohonen Feature Map (KFM). The use of unsupervised networks has many practical advantages, such as that the machining operation is not interrupted for taking measurements of wear. There is also the advantage of practical implementation if machining conditions change frequently and appropriate training samples cannot be collected for supervised learning. Furthermore, the numerous different combinations of tool and workpiece materials and geometries may make supervised learning impossible. Govekar and Grabec [7] used the SOM for classifying drill wear, where the SOM is used as an empirical modeler. They found that the adaptability of the SOM and its ability to handle noisy data made the technique feasible for online TCM. Jiaa and Dornfeld [8] used the SOM for predicting and detecting tool wear during turning.

The most commonly supervised NNs for TCM are the multilayer perceptron (MLP), recurrent neural network (RNN), supervised neuro-fuzzy system (NFS-S), time delay neural network (TDNN), single layer perceptron (SLP) and the radial basis function (RBF) network. The use of an SLP for TCM is described by Dimla et al. [9], using the perceptron learning rule. The SLP can only be used to identify discrete classes of a tool's condition. MLPs are usually trained with the backpropagation algorithm, which is the preferred choice for most standard cases. Some techniques are known to outperform backpropagation in terms of training time and generalization capabilities with certain NN paradigms. Lou and Lin [10] describe the use of an FF network using a Kalman filter to prevent the training problems encountered with backpropagation training for a TCM application. The method is less sensitive to the initialization values of the weights and biases that sometimes cause convergence problems with backpropagation.

Generally speaking, a dynamic system such as cutting processes should be monitored by means of a dynamic modeling technique, such as dynamic NN paradigms. To this end, recurrent networks, or even combining recurrent networks with other NN paradigms, can be used. Networks with tapped delay lines can also be used, such as TDNNs. This paper proposes a combined implementation of static and dynamic supervised NNs.

## 1.2 Hidden Markov models

The use of HMMs for the modeling of speech has been well established in the speech recognition community the past 20 years. HMMs were first introduced by Baum et al. [11]. After the introduction of HMMs, the method has been used successfully in speech recognition for several years. Consequently, there is a large variety of computer software available which enables one to implement HMM-based modeling easily.

HMM modeling is a statistical modeling technique which is capable of characterizing the observed data samples of a discrete-time series. Bunks et al. [5] discuss the similarities between speech processing and TCM and give reasons for the successful application of HMMs for the modeling of TCM. HMMs are an extension of Markov chains, i.e. an HMM consists of a finite set of  $N$  states that is traversed according to a set of transition probabilities. The transition probabilities describe the conditional probability that the HMM will occupy a specific state, given a history of the states previously occupied. The temporal nature is therefore modeled by the state transition probabilities. Each state has a conditional probability distribution associated with the output, which defines the conditional probability that the HMM will emit an observation symbol (or feature vector), given that the model is occupying a specific state. Therefore, unlike Markov chains, an HMM concurrently models two stochastic processes: the temporal structure and the locally stationary structure of the system. In the case of TCM, the assumption made is that the wear on tool inserts is locally stationary but does increase over time. The temporal structure is modeled by the state transition probabilities and the locally stationary character is modeled by the conditional probability density function of the output. The state sequence is described as hidden, because only the sequence of observation symbols is known (hence the name hidden Markov model). The HMM can be viewed as a doubly embedded stochastic process where the underlying stochastic process (the state sequence) is not directly observable. If we translate the HMM terminology into TCM terms, the wear of the tool inserts is represented by the hidden state sequence and the sensor measurements are the observation symbols (or feature vectors).

HMMs are typically used for classifying classes or patterns (as Bunks et al. [5] did in order to classify defects). An HMM is trained for each of the different patterns that one wants to identify and the model that best fits the test data is regarded as the most likely model to have generated the data. This paper describes using an HMM to estimate the current wear from the sensor input data. This was done by determining the optimal state sequence given the sensor input data. HMMs have been put to similar use when processing speech, for example to estimate the pitch of a voice, which information is needed for performing speech-processing tasks such as speech enhancement and speech recognition [12]. As the state sequence represents the optimal estimate of

the current tool wear, the HMM and NN modeling of tool wear could be compared.

The following notation is used to describe an HMM:

- $T$ : length of the observation sequence (sensor input data).
- $N$ : the number of states in the model.
- $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \dots, \mathbf{o}_T)$  – the observation sequence (sensor input data).
- $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots, \mathbf{q}_N)$  – the states.
- $\mathbf{S} = (s_1, s_2, s_3, \dots, s_T)$  – the hidden state sequence (representing the tool wear).
- $\mathbf{A} = [a_{ij}]$ ,  $a_{ij} = (P_{s_t = q_j | s_{t-1} = q_i})$ ,  $i, j = 1, 2, \dots, N$ —state probability transition matrix.
- $\mathbf{B} = b_j(\mathbf{o}_t)$ ,  $b_j(\mathbf{o}_t) = f(\mathbf{o}_t | s_t = q_j)$ ,  $j = 1, 2, \dots, N$ —the state output probability distributions (pdfs).
- $\pi = \pi_i$ ,  $\pi_i = (P_{s_1 = q_i})$ ,  $i = 1, 2, \dots, N$ —the initial probabilities of being in each different state.

A complete specification of an HMM,  $\Phi$ , includes three sets of probability measures  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\pi$ . For convenience an HMM is represented by the notation

$$\Phi = (\mathbf{A}, \mathbf{B}, \pi) \quad (1)$$

There are three important algorithms when using HMMs. They are stated as follows: Given a model  $\Phi$  and a sequence of observations  $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \dots, \mathbf{o}_T)$ ,

- *Evaluation problem*: What is the probability  $P(\mathbf{O}|\Phi)$ , i.e. the probability that the model generates the observations?
- *Decoding problem*: What is the most likely state sequence  $\mathbf{S} = (s_1, s_2, s_3, \dots, s_T)$  in the model that produces the observations?
- *Training problem*: How should the new model parameter  $\hat{\Phi}$  be reestimated from the current model parameters in order to maximize the joint likelihood  $\prod_{\mathbf{O}} P(\mathbf{O}|\hat{\Phi})$ ?

This paper is only concerned with the solutions to the Training and Decoding problems. The solution to the training problem is called the Baum-Welch algorithm, which produces maximum likelihood estimates of the HMM parameters. The solution to the decoding problem is called the Viterbi algorithm. As the theory of HMM modeling has been extensively covered in the literature, we do not review HMM theory in this paper. For an in-depth discussion of the background theory of HMM modeling, the reader may refer to [4, 13, 14].

## 2 Data collection

### 2.1 Experimental setup

The data for the comparative evaluations presented in this paper was obtained by means of the measurement system used in a previous study involving the monitoring of industrial tool wear [15]. Data was collected by a system that can automatically log the cutting forces

during the machining of automotive pistons, and was installed in the plant of a piston manufacturer. The measurement system consists of the following:

- A tool-holder with strain gages (3 half-bridges)
- Strain gage amplifiers
- Anti-alias filters
- A/D conversion
- A computer with data-logging software.

Figure 1 contains a diagram showing the layout of the data collection system.

The development of the measurement system and its calibration is described in [16], and therefore only basic remarks are repeated here. Special data-logging software was developed to trigger automatically the onset of data processing each time the tool engages with the workpiece. As the computer is also a web server, the variables for data collection and storage can be set remotely. In order to collect cutting forces, a calibration matrix was determined through a number of controlled loading experiments. The three incoming voltage signals consequently yield the cutting forces in the three main directions, i.e.  $F_x$ ,  $F_y$  and  $F_z$ . It was found that the system is about 99% accurate in the  $F_x$  and  $F_y$  directions, whereas the accuracy of  $F_z$  is about 85% (due to the longitudinal stiffness of the boring bar that yields low strains). Despite the limited accuracy, the system can be realized at a fraction of the cost incurred when using alternative sensors. The proposed system is also more robust than any other currently available hardware, and was found to perform well in a production environment [15].

The machining process considered for this paper is an interrupted boring operation on an automotive piston. Figure 2 is a picture of the machining operation, showing the sensor-integrated tool. The experimental conditions are summarized in Table 1. The machining process essentially involves removing excess metal at two locations inside the automotive piston.

2.2 Tool wear

Before discussing a further analysis of the force signals, some remarks should be made about the tool wear. For

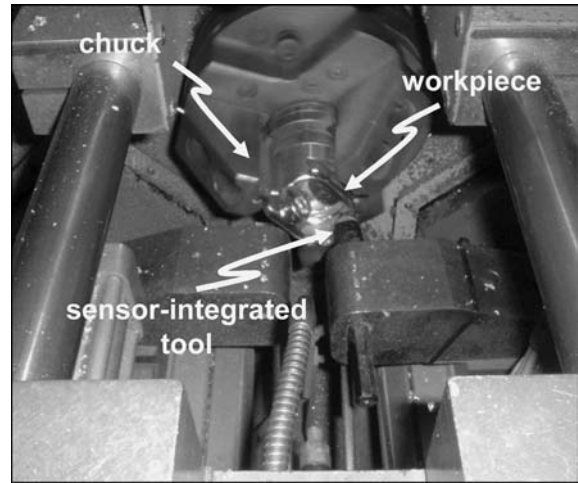


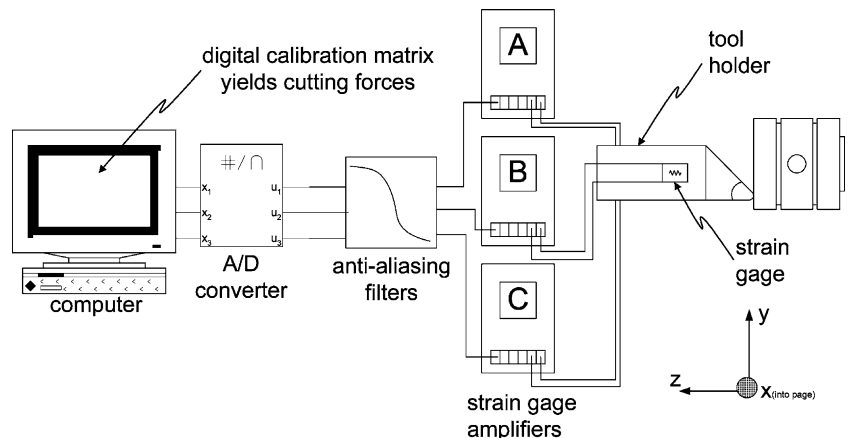
Fig. 2 Machining operation

Table 1 Experimental conditions

Machine	Feed rate	0.3 mm/rev
	Cutting Speed	390 m/min
	Depth of cut	Varies: 0–1.7 mm max
	Tool-holder	S16 QSCLPR 09
	Insert type	CCGT 09T304 FN (carbide)
	Workpiece	Aluminum alloy
Strain Gages	3 half-bridge configurations	
	Type	KYOWA KFG-1-120-D17-11
	Amplifier	HBM clip System
Filter	Custom built –3dB at 4 kHz	
A/D card	Eagle Technologies PC-30	
	Sampling Rate	20 kHz per channel
	Sampling Time	3 s per channel

all wear measurements and subsequent wear estimation, the average flank wear over a selected area of the cutting insert was chosen as a representative value of the tool condition, and this parameter is referred to as VB. Figure 3 is a scanning electron microscope (SEM)

Fig. 1 Data collection system



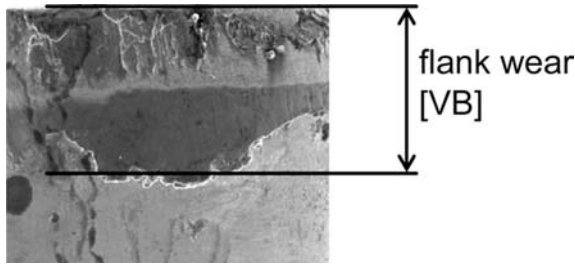


Fig. 3 Sampling electron microscope photograph of a worn tool insert

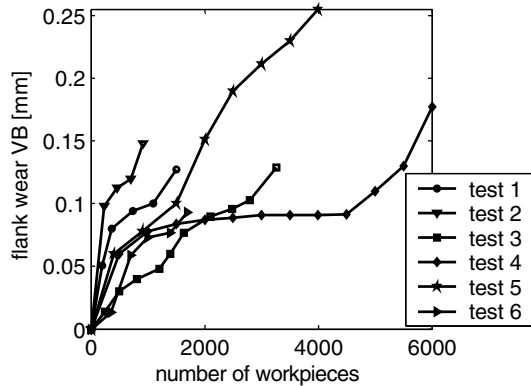


Fig. 4 Tool wear according to the number of machined workpieces

picture of a worn tool insert, also showing the value of VB for the particular case.

During the course of the research, data was collected from almost 100 tool inserts. This implies that force signals from the process were collected continuously from a new to a worn tool insert many times over. The experience of the operators on the shop floor is that the rate of tool wear is unpredictable. Sometimes a tool will last for thousands of components, and sometimes it will wear out after a few hundred. A conservative approach is then taken to eliminate the possibility of scrapping a part and tools are often replaced long before this is necessary. The unpredictability of the tool wear was confirmed by the wear measurements taken on the shop floor. Figure 4 plots a comparison of the flank wear of several tools according to the number of components that were machined with them. It can be seen that the rate of tool wear differs in each case, despite the fact that in all the cases the machining conditions remained the same.

This fluctuation in tool life can be attributed to the conditions on the shop floor. More specifically, the rate at which components are manufactured plays a significant role. If the time allowed for the tool to cool down between runs is not kept constant, large variations in tool life can be expected. Fluctuations in the workpiece composition may also play a role. A highly significant conclusion from this is that tool life equations (e.g. modified Taylor equations [17]) cannot solve the problem in this case, hence justifying the need for an online monitoring system.

### 2.3 Signal processing

In the analogue form, the sensor signals are filtered and run through an overload protection unit. Once in digital format, the signals are phase corrected, resampled, DC offset compensated and multiplied by the calibration matrix to yield the three cutting-force signals.

The rotation of the spindle was exactly 1,500 rpm (25 Hz). As described above, the removal of metal is interrupted during one revolution with two cuts per revolution, hence giving the frequency of interruption as  $2 \times 25 = 50$  Hz (not to be confused with electrical line frequency). Closer inspection revealed that the signals consisted of low-frequency and high-frequency components. The low-frequency component (50 Hz) is an indication of the static cutting forces. In the higher frequency range, the natural frequencies of the tool holder are observed because the energy from the cutting operation causes impacts that excite them. The typical time-response history of the calibrated force signals during machining is shown in Figure 5.

### 2.4 Feature selection

Several investigations were conducted to identify the possible signal features that might correlate with tool wear. A number of features were generated using

- Time domain/statistical methods (mean, standard deviation, skewness, etc.)
- Time–frequency domain methods (spectrograms, wavelet analysis)
- Frequency domain analysis (Fast Fourier Transform (FFT), Power Spectral Density (PSD), cepstrum analysis, etc.).

After having generated a list of possible features for monitoring, the most reliable features for monitoring should be selected. This is one of the most important steps in designing a monitoring system. There are

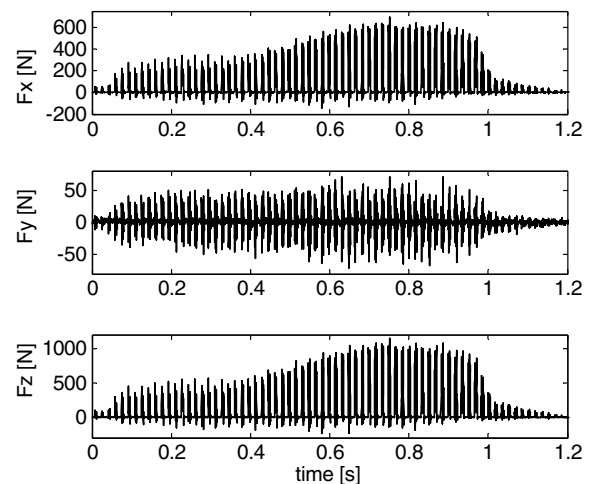


Fig. 5 Typical time-response history of cutting forces



various methods available for feature selection and feature space reduction. A discussion of such methods is beyond the scope of this paper, but in this case a simple method was employed: Since tool wear increase monotonically with time, the signal features related to tool wear also tend to increase or decrease as tool wear increases. Therefore a simple feature selection can be made by identifying the features that have a high linear correlation with the physical tool wear. The correlation coefficient (expressed as an absolute percentage) between two variables  $x$  and  $y$  is determined with:

$$r = \left| \frac{1}{n-1} \sum \left( \frac{x - \bar{x}}{\sigma_x} \right) \left( \frac{y - \bar{y}}{\sigma_y} \right) \right| \times 100 \quad (2)$$

where  $r$  is the correlation coefficient whose value indicates linearity between  $x$  and  $y$ . When  $r$  is approaching 100%, there is a relationship between  $x$  and  $y$ . The lower the value of  $r$ , the smaller the chance that the selected feature will show any trend with respect to tool wear.

As a last step of feature selection, some engineering judgement is required because the automated methods will often select features that are too similar or too dependent on one another, and therefore do not achieve the goal of sensor fusion. In this case, the rules for selecting features based on engineering judgement can be stated as follows:

- Select features from the static and dynamic parts of the signal
- Select features from the different force directions
- Use time and frequency domain features
- Features based on simple signal-processing methods are preferred
- There should be a reasonable physical explanation for the behavior of a feature with respect to tool wear.

After considering the correlations and the above-mentioned factors, the features listed in Table 2 were selected. There are several reasons for the choice of these particular features, and also for choosing only four features to monitor the process. Some scholars might argue that the features are not linearly independent and hence unsuitable for NN modeling. A discussion of these issues falls beyond the scope of this paper, but can be found in [15].

Figure 6 plots the chosen features of an interpolated vector of tool wear. Note that the features are always normalized for modeling purposes. It is clear from the figure, that although the features tend to increase as tool

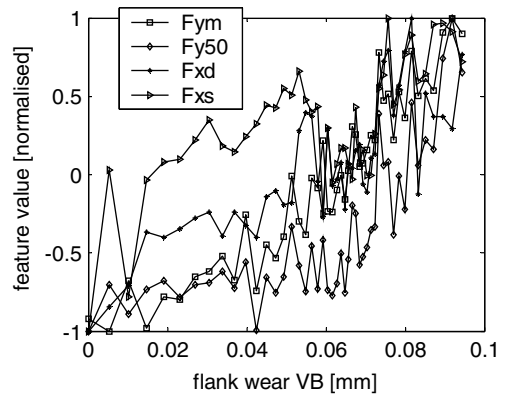


Fig. 6 Normalized features for a typical test

wear increases, the trend is extremely inconsistent. Due to the high degree of variance in the increasing trends of the features, a monitoring strategy based on any one of the features alone will never yield an accurate estimation of the tool wear. This type of problem is perfectly suited to an artificial intelligence (AI) modeling approach, hence the comparison of the two methods, NNs and HMMs, for performance on this particular data set.

### 3 Neural network

#### 3.1 Formulation

The network paradigm proposed in this paper utilizes two types of NNs: One is a dynamic network trained online and the other is a static network trained off-line. The static networks are trained to model the feature values for known values of tool wear. The dynamic network attempts to estimate the current wear on the cutting edge by using the previous estimations of tool wear as input, hence it is a type of feedback network with time delays. The training goal for the dynamic network is to minimize the error between online measurements and the output of the static networks. The approach is shown diagrammatically in Fig. 7.

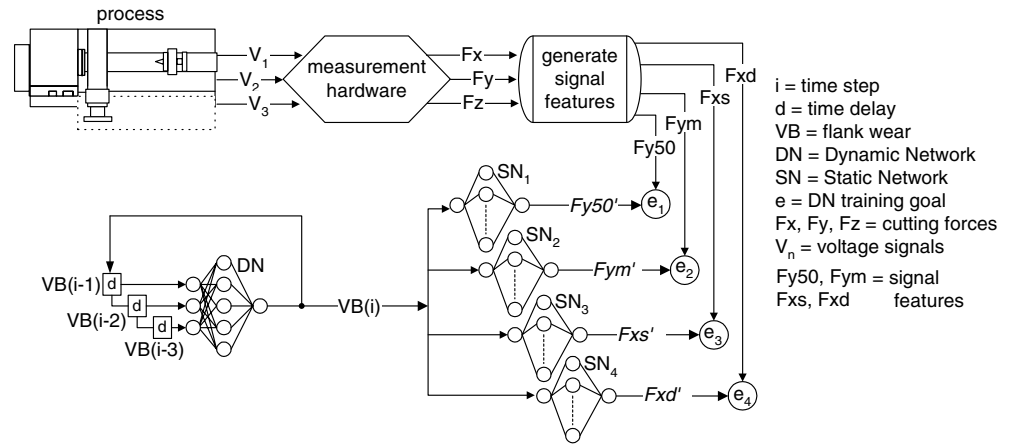
This formulation has several advantages over conventional NN paradigms. The most important advantage is the use of temporal information to estimate the next value in the time series. It might seem possible to achieve a similar result with a curve-fitting procedure instead of a dynamic NN, but the continuous growth of the tool wear was found to be too complex for such a procedure. With the proposed method, the current level of intelligence contained in the dynamic NN is combined with the knowledge from the online sensors to make the best possible decision about the severity of wear on the tool. Therefore the dynamic network can follow any geometric progression of tool wear.

The static networks were trained, validated and tested for each of the chosen features. Four relatively small static networks are used, all of which are FF networks with three layers. The middle layer consists of five

Table 2 Description of features

Feature	Description
Fxs	Standard deviation of Fx
Fxd	Spectral energy of Fx at the first natural frequency of the tool holder (approx. 2 kHz)
Fym	Mean of Fy
Fy50	Energy around 50 Hz for Fy

Fig. 7 Monitoring strategy



‘tansig’ neurons, and the output neuron has a linear activation function. The static networks were trained with Levenberg-Marquardt backpropagation. One of the main considerations when training NNs is to prevent overtraining. This will cause the networks to memorize the training data and as a result they are not able to generalize when they are presented with new data. In this case, the use of small networks was combined with early stopping of training to prevent this effect.

After training the static networks, the dynamic network is trained online to estimate the online value of the tool wear,  $VB(i)$ . One of the main reasons why this approach is so efficient for TCM applications is that tool wear seldom follows the same geometry and growth rate. If the static networks are trained appropriately, the dynamic network can follow any growth and geometry of tool wear. The dynamic NN is of the same type as the static NNs, namely an FF network with three layers. The training target of the dynamic network is to minimize the difference between the actual features from the online force measurements and the output of the static networks.

The proposed method utilizes sets of inner and outer steps or time increments. The inner steps are the training steps of the dynamic NN to achieve a specified convergence. Hence, during the inner steps, the tool wear is assumed to be constant and the NNs attempt to estimate this value. When this is achieved, an outer step is taken, and in this case it is an incremental step in the tool wear. The problem can be described by considering a vector  $\mathbf{x}$  containing the network bias and weight values of the Dynamic Network (DN):

$$\mathbf{x} = [x_1, x_2, \dots, x_n] \quad (3)$$

In order to increment an outer step, the following optimization problem has to be solved, which is the training goal of the DN:

$$\text{Minimize } f(\mathbf{x}) = \sum_{j=1}^4 e_j \quad (4)$$

such that

$$f(\mathbf{x}) \leq \text{tol} \quad (5)$$

with the initialization space for a new tool starting at:

$$D = \{(x_1 \dots x_n) \in \mathbb{R}^n : -1 \leq x_i \leq 1, \quad i = 1, \dots, n\} \quad (6)$$

and where *tol* is a suitable convergence tolerance on the function value. The initialization space for a worn tool is obtained from the solution of the previous outer step. The error functions in Equation (3) are defined as follows:

$$\begin{aligned} e_1 &= \sqrt{(Fy50' - Fy50)^2} \\ e_2 &= \sqrt{(Fym' - Fym)^2} \\ e_3 &= \sqrt{(Fxs' - Fxs)^2} \\ e_4 &= \sqrt{(Fxd' - Fxd)^2} \end{aligned} \quad (7)$$

Fy50', Fym', Fxs' and Fxd are the output of the static networks SN<sub>1</sub>, ..., 4. When the DN reaches its training goal, an outer step can be taken and new values for Fy50, Fym, Fxs and Fxd can be measured using the online system for measuring the cutting force. Note that all variables are normalized before they are entered into the NNs, and denormalized at the network output for the interpretation of the results. The Particle Swarming Optimization Algorithm [18] was used for training the DN, which yielded rapid and reliable convergence.

### 3.2 Results

Separate data sets were used for training and testing the NN strategy. The static NN training was validated with the training data set. The test data set is the data that the NN has not seen before, but is data obtained under the same experimental conditions. The test data set consists of three complete cases of tool wear, namely the case starts with a new tool which is replaced twice by a new tool after a fair amount of wear has occurred. Figure 8 shows the signal features (as discrete time steps) as they were presented to the NN strategy.

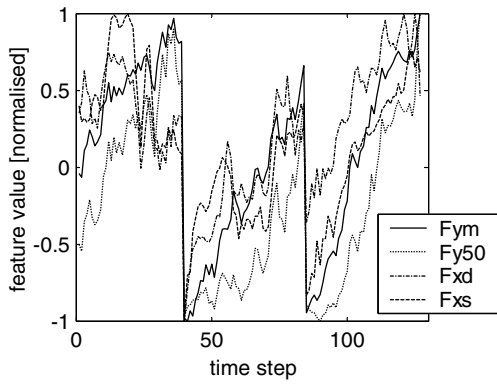


Fig. 8 Testing features

The simulation results of the NN strategy are shown in Fig. 9 in direct comparison with the actual measured value of the tool wear, which was fitted to a 3rd -order polynomial. Tool wear normally has initial, regular and fast wear stages, and it has been found that a 3rd order polynomial provides a reasonable fit to typical experimental tool wear data [19]. The figure indicates that the performance of the NN is quite good. The strategy was also tested with reinitializations and retraining of the NNs and similar results in performance were found. The size and training tolerances of the static NNs play an important role in ensuring good results. The optimal values were determined by trail and error.

## 4 Hidden markov model

### 4.1 Formulation

The NN formulation directly models the progression of tool wear  $VB(i)$  and is capable of following any progression of tool wear. In the HMM paradigm proposed in this paper, the HMM states represent the values of tool wear. As a result, the HMM will only be able to model  $N$  different values of tool wear, as there are only  $N$  HMM states. The larger the number of states, the finer the resolution of the tool wear modeling and the smoother the estimated tool wear. A unique label has to

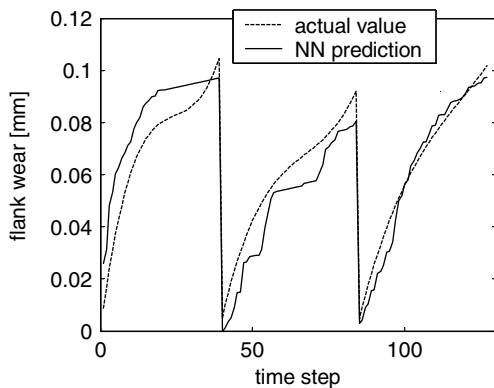


Fig. 9 Neural network simulation results

be associated with each state so that a numerical value for tool wear can be extracted. This label is a numerical value for the flank wear. The labels are determined prior to training by quantifying the expected range of tool wear. This is discussed below in more detail when discussing the choice and initialization of state output pdfs.

Different types of HMMs are suitable for modeling processes but it is important to choose the topology of the HMM and the type of state output pdfs in order to maximize the performance of HMM modeling. When deciding which type of HMM should be used for modeling the tool wear, knowledge of the general form of tool wear was utilized to guide these decisions. The HMM used for modeling the tool wear is shown in Figure 10. The choices of topology and state output pdfs are discussed in greater detail below.

#### 4.1.1 Selection and initialization of topology

The topology of an HMM is specified by the state transition matrix  $A$ . The left-to-right HMM was first proposed for the modeling of speech by Bakis [20]. The underlying state sequence associated with the left-to-right model has the property that, as time increases, the state index stays the same or increases. Therefore the HMM states can only proceed from left to right. This is ideally suited to modeling tool wear, which has the property of being a non-decreasing function over time. The state transition matrix of left-to-right HMMs is described by the transition probabilities that have the following properties:

$$\begin{aligned}
 a_{ij} &= 0, & j < i \\
 a_{ij} &= 0, & j > i + k \\
 \pi_{ij} &= \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \\
 a_{NN} &= 1, a_{Ni} = 0 & i < N
 \end{aligned} \tag{8}$$

Equation (8) states the first observation has to be assigned to the first state and therefore constrains the estimated current tool wear so that it always starts at 0 mm wear. Equation (8) constrains the HMM to allow transitions

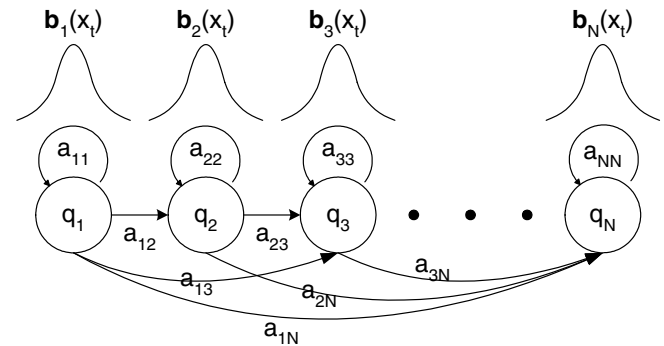


Fig. 10  $N$ -state left-to-right (Bakis) HMM



only from a state to itself or a subset of neighboring states determined by  $k$ . This property constrains the model to a limited increase in tool wear during any single state transition (or measurement). By trial and error it was determined that a choice of  $k = 50$  and  $N = 200$  gives the best results for estimating tool wear.

The transition probabilities are initialized in such a way that the self-loop transition probability is set to  $a_{ii} = 0.0075$  while the remaining transition probabilities are initialized to

$$a_{ij} = \frac{1 - a_{ii}}{L_i}, \quad i \neq j \quad (9)$$

where  $L_i$  is the number of transitions leaving state  $q_i$ .

The self-loop probabilities are set to a value less than  $1/L_i$  in order to minimize the likelihood that the estimated flank wear will contain sections of constant wear.

#### 4.1.2 Selection and Initialization of state output pdfs

The state output pdfs model the conditional probability that a specific state generated a observation sequence, given that the HMM occupies that specific state. One of the assumptions of HMMs is that successive observation features are conditionally independent. This assumption is not very accurate and in an attempt to compensate for the errors resulting from the conditionally independent assumption, dynamic features are added to pre-processed observation features. The dynamic features (also called *delta* features) are the first-order temporal derivatives of the observation features. As mentioned above, the sensor input data is regarded as the observation vectors. The pre-processed observation at time  $t$  is formed from the sensor data as follows:

$$\mathbf{o}_t^{\text{pre-processed}} = \begin{bmatrix} \text{Fy}_{50_t} \\ \text{Fym}_t \\ \text{Fxs}_t \\ \text{Fxd}_t \end{bmatrix} \quad (10)$$

The observation at time  $t$  is formed by augmenting the pre-processed observation with dynamic features as follows:

$$\mathbf{o}_t = \begin{bmatrix} \mathbf{o}_t^{\text{pre-processed}} \\ \Delta \mathbf{o}_t^{\text{pre-processed}} \end{bmatrix} \quad (11)$$

where

$$\Delta \mathbf{o}_t^{\text{pre-processed}} = -2\mathbf{o}_{t-2} - \mathbf{o}_{t-1} + \mathbf{o}_{t+1} + 2\mathbf{o}_{t+2} \quad (12)$$

and the temporal derivative have been computed from a polynomial approximation of the time derivative. The HMM therefore models the trajectory of the sensor measurements in an 8-dimensional feature space. We have chosen to use multivariate Gaussian distributions to model the input data.

The state output pdf of state  $q_j$  takes the form of

$$b_j(\mathbf{o}_t) = (2\pi)^{-\frac{D}{2}} |\Sigma_j|^{-1/2} e^{-(1/2)(\mathbf{o}_t - \mu_j)^T (\Sigma_j)^{-1} (\mathbf{o}_t - \mu_j)}, \quad (13)$$

where  $\mu_j$  and  $\Sigma_j$  and respectively the mean and covariance matrix of the Gaussian pdf.

In order to improve the condition number of the covariance matrices (and to avoid ill-conditioned matrices due to the finite precision of computers), it was necessary to normalize the pre-processed observation features. The normalization scales each dimension of the observation features so that the standard deviation of that dimension is unity. The scaling for each dimension is computed from the training data.

The resultant quality of the HMM modeling is highly dependent on the initialization of the output pdfs. Three parameters need to be initialized for each of the  $N$  states. The first two parameters are the mean and the covariance matrix of the Gaussian pdf. The third parameter is the label (the numerical value of tool wear) associated with each state. The mean of the pdf and the label of each state are initialized concurrently through the use of vector quantification. A numerical value of the tool wear is associated with each training observation. The set of 8-dimensional observation vectors  $\mathbf{O}$  is divided into  $N$  clusters using binary split vector quantification on the observation features. The mean of each of the  $N$  observation clusters is used to initialize the mean of each the  $N$  state output pdfs. However, when the  $N$  observation clusters are formed, there is an associated tool wear cluster that is formed concurrently. The mean of each of the  $N$  tool wear clusters is used as a label for each of the  $N$  states. Through the use of vector quantization we formed  $N$  pairs, consisting of the pdf mean and label for each  $N$  state. However, the mean-label pairs cannot be assigned to the states at random, because the HMM has to model the flank wear as a non-decreasing function of time. To ensure that as the index of the states increase, the estimate tool wear increases too, the mean-label pairs are sorted in ascending order of numerical tool wear and assigned as shown in Fig. 11.

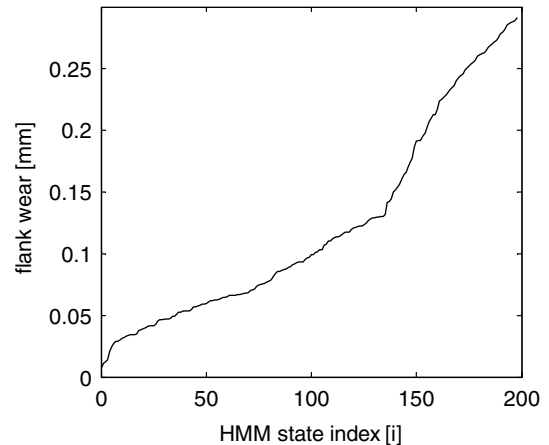


Fig. 11 Initialization of tool wear labels of HMM states

As the amount of training data was limited, a decision was taken that the covariance matrices would be constrained to be circular. The only HMM parameter that was updated during training was the state transition matrix  $\mathbf{A}$ . A circular Gaussian pdf has a covariance matrix that can be expressed as follows:

$$\Sigma_j = \sigma_j^2 \mathbf{I}, \quad \text{where } \mathbf{I} \text{ is the identity matrix} \quad (14)$$

The covariance matrices are initialized with a variance of  $\sigma_j = 0.05$ . The process of initializing the HMM state output pdfs is diagrammatically depicted in Fig. 12.

## 4.2 Results

The same data sets that were used for training and testing the NN strategy were used for evaluating the

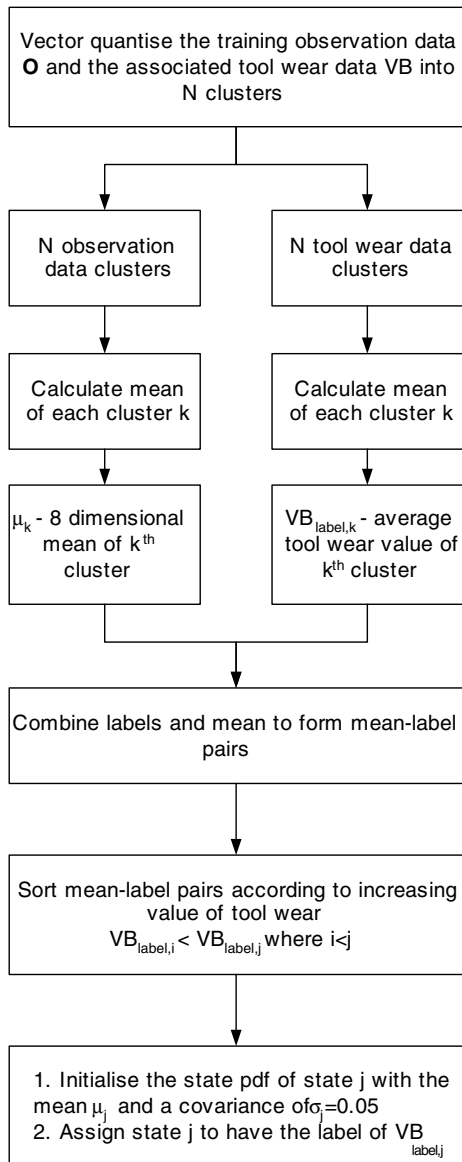


Fig. 12 Initialisation of HMM state output pdfs

HMM strategy. The training set consists of ten examples of tool wear and the test set consists of three examples. The results were determined using the Viterbi decoding algorithm, which gives the optimal tool wear, given the complete observation sequence. Therefore the implemented HMM strategy is not strictly an online estimation tool. The simulation results of the HMM strategy are shown in Fig. 13 in direct comparison with the actual measured value of the tool wear (fitted to a 3rd-order polynomial).

The above figure indicates that the HMM manages to predict the actual tool wear. Because the Baum-Welch and Viterbi algorithms will always produce the same output, given the same observation features and HMM model, it is not necessary to reinitialize and retrain the HMM as the HMM was initialized in a deterministic manner. The results are sensitive to the number of HMM states, and the self-loop transition probability and the optimal values were determined by trail and error.

## 5 Comparative evaluation

### 5.1 Results

Figure 14 shows the simulation results from both methods, compared with the data on the actual measured tool wear. Considering the noisy nature of the measured data, both methods actually performed exceptionally well in predicting the flank wear on the cutting tool.

The simulation errors, expressed as the root mean square (rms) error of the actual value of the tool wear, are summarized in Table 3 for each of the three test cases from a new to a worn tool. The average rms errors for the two methods are nearly identical.

The  $\chi^2$  test is often used to determine if mathematical models associate with experimental data. It is calculated with

$$\begin{aligned} \chi^2 &= \frac{(z_1 - E_1)^2}{E_1} + \frac{(z_2 - E_2)^2}{E_2} + \dots + \frac{(z_k - E_k)^2}{E_k} \\ &= \sum_{i=1}^k \frac{(z_i - E_i)^2}{E_i} \end{aligned} \quad (15)$$

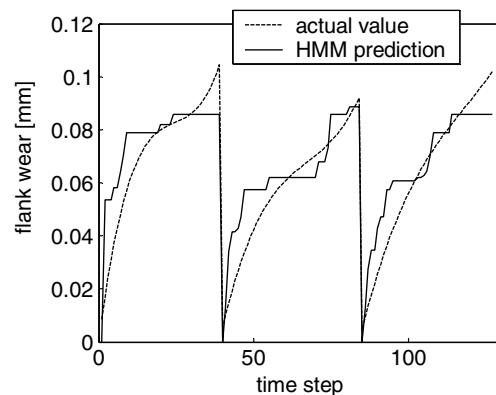


Fig. 13 HMM simulation results

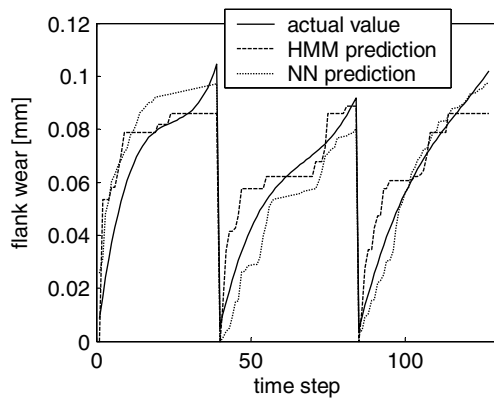


Fig. 14 Final simulation results

Table 3 Simulation rms errors

	Root mean square error (mm)	
	Neural network	Hidden Markov model
Test 1	0.0117	0.0091
Test 2	0.0095	0.0078
Test 3	0.0040	0.0078
Average	0.0083	0.0082

where  $z_i$  are model predictions and  $E_i$  are the actual (experimental) values. In this case, the  $\chi^2$  test can be used to determine which of the two methods yielded the best fit to the test data. Table 4 shows the results of  $\chi^2$  tests between the model predictions and experimental values. According to the  $\chi^2$  test, the NN outperforms the HMM by a slight margin. Based on these results, it can be concluded that the NN (with optimal network size, type and training) is the better technique for continuous estimations.

## 5.2 Discussion

It is clear from the results presented above that both approaches yield exceptional results in predicting the tool wear. The advantage of the NN is its ability to perform continuous estimations of the tool wear (hence it predicts the future value of the tool wear when presented with historical data). The HMM does not

Table 4  $\chi^2$  test results

	$\chi^2$	
	Neural network	Hidden Markov model
Test 1	0.1678	0.2026
Test 2	0.1188	0.1536
Test 3	0.0340	0.1242
Average	0.1068	0.1601

perform continuous estimations: It is presented with the complete set of features for an experiment before it makes a prediction. However, the current formulation of the HMM can be altered to enable continuous estimations, in which case it might yield slightly less accurate estimations. A conclusive answer to the question of whether NNs or HMMs are better suited to the continuous estimation of tool wear would require the collection of more data for training and testing purposes.

The disadvantage of the NN is that it is more complex than HMMs. Selecting the correct network structure to solve a particular problem can be quite difficult with NNs and often this requires substantial trial and error simulations, or will require automated optimization. Furthermore, selecting the correct training algorithm and convergence criteria with NNs also requires trial and error work. This makes it difficult to implement NNs without the relevant experience. Although training time is often cited as one of the disadvantages of NNs, it is rarely a real problem as computational power has become cheap and fast. In this case, training could be done in a matter of seconds.

The advantage to using HMMs is that they are exceptionally easy to train and test, once the HMM has been initialized. The Baum-Welch and Viterbi algorithms are the standard algorithms used for training and decoding HMMs. The Baum-Welch algorithm is guaranteed to converge to a local optimum, and therefore correct training depends on the initialization of the HMM. The choices of topology and state output pdfs are largely dependent on the knowledge and understanding of the problem and the amount of training data available. The disadvantage of HMMs is that they generally have a large number of parameters and therefore a large amount of training data is necessary to ensure that the HMM parameters are well estimated. Furthermore, first-order HMMs generally do not adequately model the long-term temporal behavior of processes, owing to the first-order Markov assumption [4]. It is sometimes necessary to model explicitly the state time duration [21, 22] or to use higher-order HMMs in order to improve the temporal modeling. The problem with higher-order HMMs is that they require far more training data for successful implementation. Lastly, HMMs are generally used as a classification tool rather than a tool for continuous estimation. Although it is theorized that the HMM can be modified to be a continuous estimation tool, this has not been tested.

With all the results taken into account, it can be said that both methods performed well in an application for monitoring tool wear and it is likely that both will also yield comparable results for many other industrial applications.

## 6 Conclusion

Monitoring of the wear on the tools of machines used in manufacturing applications has many advantages for

the end-users of such machines. The topic of sensor-based monitoring of tool wear has been widely researched for a variety of different machining operations such as turning, drilling and milling. Neural networks are frequently used for classifying the features extracted from sensor data. These classifications are normally attempts to make continuous estimations of the tool wear. Recently, HMMs have also been employed for the same purpose. The aim of this paper was to compare simulations directly with NNs and HMMs in an application for monitoring tool wear. Force measurements were taken from a boring operation in a mass-production plant. Signal features were extracted from the data that correlated with progressive wear on the tool. Despite the noisy nature of the data, the features could be successfully used with NNs and HMMs to monitor the wear on the tools. Both methods performed well, with NNs providing a slightly better fit to the test data according to the  $\chi^2$  test results. Thus, based on the simulation prediction errors alone, it is not possible to indicate a particular preference for either of the two methods.

The advantage of NNs is their ability to perform continuous estimations (although HMMs can also be employed to achieve this). The disadvantages of NNs are their relative complexity compared with HMMs and also the fact that a great deal of experience and trial-and-error work are required for successfully implementing an NN-based monitoring strategy. The advantages of HMMs are that if the problem is well understood it is fairly easy to initialize and implement an HMM-based monitoring strategy, since computer implementations of HMMs are readily available. However, HMMs typically contain a large number of parameters and therefore need large amounts of data to estimate the HMM parameters properly. Lastly, HMMs are not generally used for making continuous estimations, but rather for carrying out classification tasks.

The data used in this paper to compare the two methods should represent a typical industrial situation where the obtained data sets are often noisy and incomplete. Future direct comparisons of different types of data sets should be made in order to determine if either method is better suited to certain types of problems.

---

## References

1. Teti R (1995) A review of tool condition monitoring literature database. *Ann CIRP* 44:659–667
2. Sick B (2002) Online and indirect tool wear monitoring in turning with artificial neural networks: a review of more than a decade of research. *Mech Syst Signal Processing* 16:487–546
3. Ertunc HM, Loparo KA (2001) A decision fusion algorithm for tool wear condition monitoring in drilling. *Mech Syst Signal Processing* 41:1347–1362
4. Ertunc HM, Loparo KA, Ocak H (2001) Tool wear condition monitoring in drilling operations using hidden Markov models (HMMs). *Mech Syst Signal Processing* 41:1363–1384
5. Bunks C, McCarthy D, Al-Ani T (2000) Condition-based maintenance of machines using Hidden Markov Models. *Mech Syst Signal Processing* 14:597–612
6. Silva RG, Rueben RL, Baker KJ, Wilcox SJ (1998) Tool wear monitoring of turning operations by neural network and expert system classification of a feature set generated from multiple sensors. *Mech Syst Signal Processing* 12:319–332
7. Govekar E, Grabec I (1994) Self-organizing neural network application to drill wear classification. *Trans ASME: J Eng Indust* 116:233–238
8. Jiaa CL, Dornfeld DA (1998) A self-organizing approach to the prediction and detection of tool wear. *ISA Trans* 37:239–255
9. Dimla DE, Lister PM, Leighton NJ (1996) Investigation of a single-layer perceptron neural network to tool wear inception in a metal turning process. In: *Proceedings of the 1997 IEEE colloquium on modelling and signal processing for fault diagnosis*, pp 3/1–3/4
10. Lou K, Lin C (1997) An intelligent sensor fusion system for tool monitoring on a machining centre. *Int J Adv Manufacturing Technol* 13:556–565
11. Baum LE, Eagon JA (1967) An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull Am Math Soc* 73:360–363
12. Wu M, Wang DL, Brown GJ, A multi-pitch tracking algorithm for noisy speech, *IEEE Trans. on Speech and Audio Processing* 11:229–241
13. Rabiner LR (1989) A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc IEEE* 77(2):257–286
14. Rabiner LR, Juang BH (1986) An introduction to hidden Markov models. *IEEE ASSP Mag* 1(3):4–16
15. Scheffer C, Heyns PS (2004) An industrial tool wear monitoring system for interrupted turning. *Mech Syst Signal Processing* 18:1219–1242
16. Scheffer C, Heyns PS (2002) A robust and cost-effective system for conducting cutting experiments in a production environment. In: *Proceedings of 3rd CIRP international conference on intelligent computation in manufacturing engineering (ICME 2002)*, Ischia, Italy, 3–5 July 2002, p 329–334
17. Jawahir IS, Li PX, Gosh R, Exner EL (1995) A new parametric approach for the assessment of comprehensive tool wear in coated grooved tools. *Annals of the CIRP* 44:49–54
18. Kennedy J, Eberhart RC (1995) Particle swarm optimisation, *Proceedings of the 1995 IEEE international conference on neural networks*, vol 4. Perth, Australia, pp 1942–1948
19. Scheffer C (2002) Development of a wear monitoring system for turning tools using artificial intelligence. PhD Thesis, University of Pretoria
20. Bakis R (1976) Continuous speech word recognition via centisecond acoustic states, In: *Proceedings of 91st meeting of the Acoustic Society of America (Washington, DC)*
21. Levinson SE (1986) Continuously variable duration Hidden Markov model for automatic speech recognition. *Comput Speech Lang* 10:29–45
22. Russell MJ, Moore RK (1985) Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In: *Proceedings on international conference on Acoustic, Speech and Signal Processing*, Tampa, 26–29 March 1985, p 5–8