BAYESIAN SUPPORT VECTOR REGRESSION FOR TOOL CONDITION MONITORING AND FEATURE SELECTION

Jianfei Dong
Department of Mechanical Engineering, National University of Singapore, Singapore 119260
g0202086@nus.edu.sg

Geok Soon Hong
Department of Mechanical Engineering, National University of Singapore, Singapore 119260
mpehgs@nus.edu.sg

Yoke San Wong
Department of Mechanical Engineering, National University of Singapore, Singapore 119260
mpewys@nus.edu.sg

ABSTRACT – This paper introduces the application of Bayesian support vector regression (SVR) and automatic relevance determination (ARD) methods for the selection of relevant features derived from force signal for tool condition monitoring (TCM) during face milling processes.

7 primary features used by other researchers are considered, including the power spectral density, skewness, kurtosis, average and maximum force, root mean square of force, and the residual error based on the AR^1 model. A two-step approach is applied to extract the features. In the first step, the 7 primary features are derived. And then a moving window is used to calculate the mean and variance value of each primary feature. As a result, 14 features are obtained and fed into the ARD model. Different features have been found to be sensitive to two different phenomena, micro-chipping and gradual wear. The selected features from all the experiments are combined together to make them applicable for different cases.

An additional set of experimental data is used to test the generalization capability of the features. The comparison between the selected features and the rejected ones prove that the selected features are really more useful. Finally, a moving average approach is proposed to further process the regression results. And fairly good estimation result has been achieved using the selected features.

1. INTRODUCTION

It is widely acknowledged that tool condition monitoring systems are very important to realize the fully controlled machines. Many different approaches have already been developed by a lot of researchers. The early work was focused on the time-series analysis methods. Mohri et al. [1] used a very high order autoregressive model to detect tool breakage. Altintas [2] indicated that the high order model was impractical for online implementations due to the burden in computing, and presented an AR1 model to do the job. The basic idea behind these is to consider the theoretical force variation characteristic of milling signals. Other researchers began to use neural networks, as the theories became more and more popular. Tarng et al. [3] used an MLP to sense tool breakage. Tansel et al. [4] evaluated both restricted Coulomb energy (RCE) and adaptive resonance theory (ART2)-type neural networks. In both studies, indictors of tool failure were used as inputs to the network. Tarng et al. used the variable cutting force, obtained by subtracting the median cutting force from the resultant average cutting force. Tansel et al. represented the force profile by 10 values from one single tool rotation through averaging the cutting force readings every 36 degree. Kim [5] asserted that the mean, maximum, and root mean square values of the force signal indicated the tool state very well. In addition, power spectral density, skewness, and kurtosis have been successfully used as features for monitoring turning processes by many researchers, e.g. Niu et al [6]. As a result, we think it may also be a good practice to investigate them in milling processes.

All of the features mentioned above have been proved useful to represent tool conditions. However, few efforts have been made to compare them. Another factor pushing us to do such a study is that the smaller the feature dimension, the less the time needed to get the output of the neural network. Besides, instead of just monitoring the status of milling inserts, we attempt to estimate wear values in the study. For both the purposes of feature selection and tool wear estimation, we find that the approach of Bayesian support vector regression works very well.

Support vector machines (SVM) for regression (SVR), as described by Vapnik [7], exploit the idea of mapping input data into a high dimensional (often infinite) reproducing kernel Hilbert space (RKHS). The sophisticated relationship between the input and output data may have a potential to represent the complicated relationships between the wear value and its indicators. In addition, the SVR methods have many other advantages, including a global minimum solution as the minimization of a convex programming problem, relatively fast training speed, and sparseness in solution

representation. However, as pointed out by Tipping [8], the traditional SVM methodology also exhibits significant disadvantages, e.g. it cannot produce probabilistic predictions. The application of Bayesian approaches to neural networks, originated by Buntine and Weigend [9], MacKay [10] and Neal [11], can solve this problem effectively. According to Mackay [10], Bayesian probability theory provides a unifying framework for data modeling which offers several benefits, such as solving the over-fitting problem and handling uncertainty in a natural manner. As a result, Bayesian support vector regression method, which combines SVR and Bayesian approaches together, may act as a more powerful estimation tool.

Based on the Bayesian approaches, MacKay and Neal proposed a new method, called automatic relevance determination (ARD), which has the advantages of lower computational cost and better performance. The aim of ARD is to automatically determine which of many inputs to a neural network are relevant to prediction of the targets. This is done by making the weights on the connections out of each input unit have a distribution that is controlled by a hyperparameter associated with that input, allowing the relevance of each input to be determined automatically as the values of these hyperparameters adapt to the data (Neal [11]). However, the original work by the two researchers was mainly based on multilayer perceptrons (MLP).

According to Williams [12], ARD methods could be directly embedded into the covariance function between the outputs corresponding to inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ as follows:

$$Cov\left[f(\mathbf{x}_i), f(\mathbf{x}_j)\right] = Cov(\mathbf{x}_i, \mathbf{x}_j) = k_0 \exp\left(-\frac{1}{2}\sum_{i=1}^{d} k_l\left(\mathbf{x}_i^l - \mathbf{x}_j^l\right)^2\right) + k_b \quad (1)$$

where k_0>0 denotes the average power of $f(\mathbf{x})$; k_l >0 is the ARD parameter that determines the relevance of the *l-th* input dimension to the prediction of the output variables; k_b>0 denotes the variance of the offset to the function $f(\mathbf{x})$; and $\mathbf{x}^l$ denotes the *l-th* element of the input vector $\mathbf{x}$. Note that the expression is the same as the Gaussian kernel function of SVR. Therefore, this idea can be used to incorporate the ARD approach into SVR.

This paper is organized as follows: SVR algorithm is given in the next section; experimental setup and feature extraction approaches are discussed in the third section; results are presented in the fourth section; and finally we summarize the findings with a conclusion.

2. SUPPORT VECTOR REGRESSION

2.1 Bayesian Framework for SVR. Chu et al. [13] proposed a generalized framework for Bayesian SVR. We implement this framework by using a quadratic loss function. There are two reasons for choosing this loss function. One is because it can lead to an analytical expression of the network output; the other reason is that it is differentiable up to the second order, which is an essential premise to do the model selection.

In regression problems, a set of training data $\mathbf{D} = \left\{(\mathbf{x}_i, y_i) \middle| i = 1, \ldots, n, \mathbf{x}_i \in \mathbf{R}^d, y_i \in \mathbf{R}\right\}$ is collected by randomly sampling a function *f*, defined on $\boldsymbol{R}^d$. As the measurements are usually corrupted by noise, training samples can be represented as

$$y_i = f(\mathbf{x}_i) + \delta_i \qquad i = 1,2,\ldots,n \quad (2)$$

where δ_i is a Gaussian noise, given by:

$$p(\delta_i) = \frac{1}{Z_s}\exp(-C \cdot l(\delta_i)) \quad (3)$$

where Z_s equals $\int \exp(-C \cdot l(\delta_i))d\delta_i$, C is a parameter greater than zero, and $l(\delta_i)$ is the loss function, which has the quadratic form:

$$(y_i - f(\mathbf{x}_i))^2 \quad (4)$$

Thus we have $Z_s = \int \exp\left(-C \cdot \delta_i^2\right)d\delta_i = \sqrt{\pi / C}$.

The regression aims to infer the function *f*, or an estimate of it, from the finite data set ***D***. In the Bayesian approach, we regard the function *f* as the realization of a random field with a known prior probability. The posterior probability of *f* given the training data ***D*** can then be derived by Bayes' theorem:

$$P(\mathbf{f}|\mathbf{D}) = \frac{P(\mathbf{D}|\mathbf{f})P(\mathbf{f})}{P(\mathbf{D})} \quad (5)$$

where $\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)]^T$. $P(\mathbf{f})$ is the prior probability of the random field and $P(\mathbf{D}|\mathbf{f})$ is the conditional probability of the data ***D*** given the function values $\mathbf{f}$, which is exactly $\prod_{i=1}^{n} P(y_i|f(\mathbf{x}_i))$. Now the standard Gaussian processes (Williams [12]) can be used to describe a Bayesian framework.

We assume that the collection of training data is the realization of random variables $f(\mathbf{x}_i)$ in a zero mean stationary Gaussian process indexed by $\boldsymbol{x}_i$ (it is applicable to the tool wear data, when the average wear value is subtracted from each wear value). The Gaussian process is specified by the covariance matrix given in equation (1). Thus, the prior probability of the functions is a multivariate Gaussian with zero mean and covariance matrix as

$$P(\mathbf{f}) = \frac{1}{Z_f}\exp\left(-\frac{1}{2}\mathbf{f}^T\boldsymbol{\Sigma}^{-1}\mathbf{f}\right) \tag{6}$$

where $Z_f = (2\pi)^{n/2}\sqrt{|\boldsymbol{\Sigma}|}$ and $\boldsymbol{\Sigma}$ is a $n \times n$ covariance matrix given by (1).

The probability $P(\mathbf{D}|\mathbf{f})$, known as likelihood, is a model of the noise, which can be evaluated by

$$P(\mathbf{D}|\mathbf{f}) = \prod_{i=1}^{n} P(y_i - f(\mathbf{x}_i)) = \prod_{i=1}^{n} P(\delta_i) \tag{7}$$

Introducing (3), (4) into (7), we get

$$P(\mathbf{D}|\mathbf{f}) \propto \exp\left(-C\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2\right) \tag{8}$$

Based on Bayes' theorem (5), prior probability (6), and the likelihood (8), the posterior probability of $\mathbf{f}$ can be written as (according to [10], $P(\boldsymbol{D})$ is commonly ignored)

$$P(\mathbf{f}|\mathbf{D}) = \frac{1}{Z}\exp(-S(\mathbf{f})) \tag{9}$$

where $S(\mathbf{f}) = C\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 + \frac{1}{2}\mathbf{f}^T\boldsymbol{\Sigma}^{-1}\mathbf{f}$ and $Z = \int \exp(-S(\mathbf{f}))d\mathbf{f}$. The maximum a posteriori (MAP) estimation of the function values is therefore the minimization of the following optimization problem:

$$\min_{\mathbf{f}} S(\mathbf{f}) = C\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 + \frac{1}{2}\mathbf{f}^T\boldsymbol{\Sigma}^{-1}\mathbf{f} \tag{10}$$

Let $\mathbf{f}_{MP}$ denote the optimal solution of (10). Then the derivative of $S(\mathbf{f})$ w.r.t. $\mathbf{f}$ should be zero at $\mathbf{f}_{MP}$, i.e.

$$\frac{\partial S(\mathbf{f})}{\partial \mathbf{f}}\Big|_{\mathbf{f}_{MP}} = C\sum_{i=1}^{n}\frac{\partial(y_i - f(\mathbf{x}_i))^2}{\partial \mathbf{f}}\Big|_{\mathbf{f}_{MP}} + \boldsymbol{\Sigma}^{-1}\mathbf{f} = 0 \tag{11}$$

If we define $\omega_i = -C\frac{\partial(y_i - f(\mathbf{x}_i))^2}{\partial f(\mathbf{x}_i)}\Big|_{f(\mathbf{x}_i)=f(\mathbf{x}_{MP})} \forall i$

and let $\boldsymbol{\omega}$ be the column vector formed by ω_i. Then $\mathbf{f}_{MP}$ can be written as

$$\mathbf{f}_{MP} = \boldsymbol{\Sigma}\cdot\boldsymbol{\omega} = (\mathbf{I} + 2C\boldsymbol{\Sigma})^{-1}2C\boldsymbol{\Sigma}\mathbf{Y} \tag{12}$$

where $\mathbf{Y}$ stands for the column vector formed by y_i. (12) can also be decomposed into the form

$$\begin{aligned} f_{MP}(\mathbf{x}) &= \sum_{i=1}^{n}\omega_i k_0 K(\mathbf{x}, \mathbf{x}_i) + k_b\sum_{i=1}^{n}\omega_i \\ &= k_0\sum_{i=1}^{n}\omega_i K(\mathbf{x}, \mathbf{x}_i) + b \end{aligned} \tag{13}$$

where $b = k_b\sum_{i=1}^{n}\omega_i$ and $K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2}\sum_{l=1}^{d}k_l(\mathbf{x}^l - \mathbf{x}_i^l)^2\right)$ is just the Gaussian kernel in classical SVR.

2.2 Model Adaptation and Feature Selection. Let $\boldsymbol{\theta}$ be the hyperparameter vector containing the parameters in the prior distribution and the likelihood function, i.e. $\boldsymbol{\theta} = \{k_0, k_1, k_2, \ldots, k_d, k_b, C\}$. The optimal values of hyperparameters $\boldsymbol{\theta}$ can be inferred by maximizing the posterior probability $P(\boldsymbol{\theta}|\mathbf{D}) = \frac{P(\mathbf{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathbf{D})}$. As we typically have little idea of suitable values of $\boldsymbol{\theta}$ before training data are available, we assume a flat distribution for $P(\boldsymbol{\theta})$, i.e., $P(\boldsymbol{\theta})$ is greatly insensitive to the values of $\boldsymbol{\theta}$. Therefore, the evidence $P(\boldsymbol{D}|\boldsymbol{\theta})$ can be used to assign a preference to alternative values of the hyperparameters $\boldsymbol{\theta}$ (MacKay [14]),

$$\begin{aligned} p(\mathbf{D}|\boldsymbol{\theta}) &= Z_f^{-1}Z_s^{-n}\int p(\mathbf{D}|\mathbf{f}, \boldsymbol{\theta})p(\mathbf{f}|\boldsymbol{\theta})d\mathbf{f} \\ &= Z_f^{-1}Z_s^{-n}\int \exp(-S(\mathbf{f}))d\mathbf{f} \end{aligned} \tag{14}$$

An explicit expression of the evidence $P(\boldsymbol{D}|\boldsymbol{\theta})$ can be obtained from an integral over the f-space with a Taylor expansion at $\mathbf{f}_{MP}$ (where $\frac{\partial S(\mathbf{f})}{\partial \mathbf{f}}\Big|_{\mathbf{f}=\mathbf{f}_{MP}} = 0$):

$$\begin{aligned} S(\mathbf{f}) &\approx S(\mathbf{f}_{MP}) + \frac{1}{2}(\mathbf{f} - \mathbf{f}_{MP})^T\frac{\partial^2 S(\mathbf{f})}{\partial \mathbf{f}\partial \mathbf{f}^T}|_{\mathbf{f}=\mathbf{f}_{MP}}(\mathbf{f} - \mathbf{f}_{MP}) \\ &= S(\mathbf{f}_{MP}) + \frac{1}{2}(\mathbf{f} - \mathbf{f}_{MP})^T(2C\mathbf{f} + \mathbf{f}^{-1})(\mathbf{f} - \mathbf{f}_{MP}) \end{aligned} \tag{15}$$

$$\begin{aligned} p(\mathbf{D}|\boldsymbol{\theta}) &= Z_f^{-1}Z_s^{-n}\int \exp(-S(\mathbf{f}))d\mathbf{f} \\ &\propto |\mathbf{I} + 2C\boldsymbol{\Sigma}|^{-1/2}\exp[-S(\mathbf{f}_{MP})]\left(\frac{C}{\pi}\right)^{n/2} \end{aligned} \tag{16}$$

Gradient-based optimization methods are used to infer the optimal hyperparameters that maximize this evidence function, which is equivalent to minimizing the minus log of the function,

$$-\ln p(\mathbf{D}|\boldsymbol{\theta}) = \frac{1}{2}\ln\left|\mathbf{I} + 2C\boldsymbol{\Sigma}\right| + C\sum_{i=1}^{n}\left[y_i - \mathbf{f}_{MP}(\mathbf{x}_i)\right]^2 + \frac{1}{2}\mathbf{f}_{MP}^T\boldsymbol{\Sigma}^{-1}\mathbf{f}_{MP} - \frac{n}{2}\ln\left(\frac{C}{\pi}\right) \quad (17)$$

The gradients are given by:

$$\frac{\partial\left(-\ln\left(p(\mathbf{D}|\boldsymbol{\theta})\right)\right)}{\partial C} = \frac{1}{2C}trace\left[\left(\frac{1}{2C}\mathbf{I} + \boldsymbol{\Sigma}\right)^{-1}\cdot\boldsymbol{\Sigma}\right] + \sum_{i=1}^{n}\left[y_i - \mathbf{f}_{MP}(\mathbf{x}_i)\right]^2 - \frac{n}{2C} \quad (18)$$

$$\frac{\partial\left(-\ln\left(p(\mathbf{D}|\boldsymbol{\theta})\right)\right)}{\partial k_0} = \frac{1}{2}trace\left[\left(\frac{1}{2C}\mathbf{I} + \boldsymbol{\Sigma}\right)^{-1}\cdot\frac{\partial\boldsymbol{\Sigma}}{\partial k_0}\right] + \frac{1}{2}\mathbf{f}_{MP}^T\frac{\partial\boldsymbol{\Sigma}^{-1}}{\partial k_0}\mathbf{f}_{MP} \quad (19)$$

$$\frac{\partial\left(-\ln\left(p(\mathbf{D}|\boldsymbol{\theta})\right)\right)}{\partial k_j} = \frac{1}{2}trace\left[\left(\frac{1}{2C}\mathbf{I} + \boldsymbol{\Sigma}\right)^{-1}\cdot\frac{\partial\boldsymbol{\Sigma}}{\partial k_j}\right] + \frac{1}{2}\mathbf{f}_{MP}^T\frac{\partial\boldsymbol{\Sigma}^{-1}}{\partial k_j}\mathbf{f}_{MP} \quad (20)$$

where $\frac{\partial\boldsymbol{\Sigma}^{-1}}{\partial k_j} = -\boldsymbol{\Sigma}^{-1}\frac{\partial\boldsymbol{\Sigma}}{\partial k_j}\boldsymbol{\Sigma}^{-1}$.

Based on the algorithm above, the SVR and feature selection can be conducted as the following:

1) Assume an initial hyperparameter set $\boldsymbol{\theta}$.
2) Use the MAP methods to get $\mathbf{f}_{MP}$.
3) Use the gradient-based optimization methods to optimize parameters; the gradients are given by equations (18) to (20).
4) If the sum square error given by $(\mathbf{Y} - \mathbf{f}_{MP})^T \cdot (\mathbf{Y} - \mathbf{f}_{MP})$ is smaller than the predetermined threshold, then end the iteration; else return to the second step.
5) For the function values of the points to be estimated, equation (13) is used.
6) Compare the magnitude of the parameter controlling each dimension of the input data with another threshold to select the relevant dimensions.

3. EXPERIMENTAL SETUP AND FEATURE EXTRACTION

3.1 Experimental Setup. The experimental scheme for the condition monitoring system of face milling is illustrated in Figure 1 and its components are listed in Table 1.

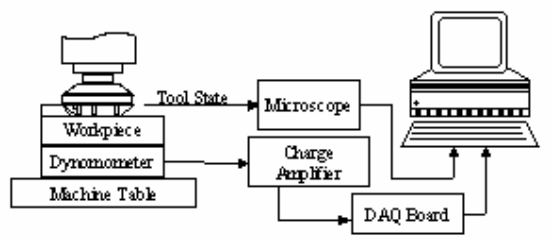


Figure 1. Experiment Setup.

Components
Makino CNC milling machine with Funuc controller
EGD 4450R cutter with AC325 inserts (Catalog No. , SDKN42MT)
ASSAB718HH workpiece
Kistler 9265B Quartz 3-Component Dynamometer
Kistler 5019A Multi-channel Charge Amplifier
NI-DAQ PCI 1200 Board
Olympus microscope and Panasonic digital camera

Table 1. Experimental Components.

The cutting force on the direction normal to feeding was captured by the Kistler dynamometer in the form of charges, which were converted to voltages by the Kistler charge amplifier. The voltage signal was sampled by the PCI 1200 board. The sampling rate was set to 1000Hz. The flank wear of each individual tooth was measured at an interval of 5 tool passes by the Olympus microscope, and at each time an average was taken from all the individual teeth mounted on the cutter. The tool state was observed by the Panasonic digital camera. Seven experiments were conducted using AC325 inserts on the Makino CNC machine. The cutting conditions are listed in the following table.

Test No.	Spindle speed (rpm)	Feed rate (mm/min)	Cut depth (mm)	Insert No.	Immersion rate
1	750	300	1	4	Full
2	1000	200	2	4	Full
3	1200	100	1	2	Half
4	1000	200	1	2	Full
5	1000	200	1	2	Full
6	900	200	1	2	Full
7*	900	100	1	2	Full

Table 2. Cutting Conditions.
(Test 7 is used to test the generalization capacity)*

3.2 Feature Extraction. A two-step feature extraction procedure was employed in the paper. The force samples in a tool revolution were first averaged to eliminate the influence of force pulsations between two successive tooth periods. The reason for not averaging the force signal in every tooth period is that the features

extracted from such average forces are quite sensitive to radial and axial run-out. Then the residual error of every force value was calculated based on the AR1 model [2]. This set of data together with the average forces formed the basis of feature extraction.

The first step can be shown as the following expression

$$P'_k(i) = \Phi_k(F(i)) \tag{21}$$

where *F(i)* stands for the i[th] set of average forces or residual errors, which are determined by a moving window (MW): $i \to i + MW - 1$. Φ_k is the mapping function; $P'_k(i)$ stands for primary features derived from mapping function; k is the k[th] type of extracted features, i.e. power spectral density, skewness, kurtosis, average, maximum, root mean square of average forces, and the moving average of residual errors.

Then the primary features were normalized with respect to an initial set of feature samples (fresh stage). This process is considered to be able to make the features less sensitive to the cutting conditions. The normalization can be expressed as

$$P_k(i) = P'_k(i) / \Lambda_k \tag{22}$$

where $\Lambda_k = \sum_{j=1}^{m} P'_k(j)/m$ is the average of the first m instances of the k[th] primary feature. Although the trends of the above primary features correlate well to tool flank wear, they cannot be reliably used due to the severe variation of the features even after the normalization preprocessing. Therefore, further processing, or the second-step feature extraction, is needed for effective tool wear estimation.

For each normalized primary feature, two secondary features were extracted, the mean and standard deviation value within a moving window *s*, which can be written as

$$mean = \frac{1}{s}\sum_{i=q}^{q+s-1} P_k(i) \tag{23}$$

$$std = \sqrt{\frac{1}{s-1}\sum_{i=q}^{q+1-s}(P_k(i) - mean)^2} \tag{24}$$

In addition, as the magnitudes of the features differ greatly, e.g. the largest one is several hundred times larger than the smallest one, scales should be used to change all of these magnitudes to comparable levels. In this case, the same level of 6 was used. The scale for each feature dimension was given by the ratio between 6 and its magnitude.

The two-step feature extraction results in 14 features, $\boldsymbol{x_i}=\{mp_i, sp_i, ms_i, ss_i, mk_i, sk_i, mm_i, sm_i, mx_i, sx_i, mr_i, sr_i, mre_i, sre_i\}$, respectively representing the mean and standard deviation value of the power spectral density, skewness, kurtosis, average, maximum, and root mean square of the average force and the average of the residual error. Figure 2 gives an example of these features, which are extracted from the data of Test 1.

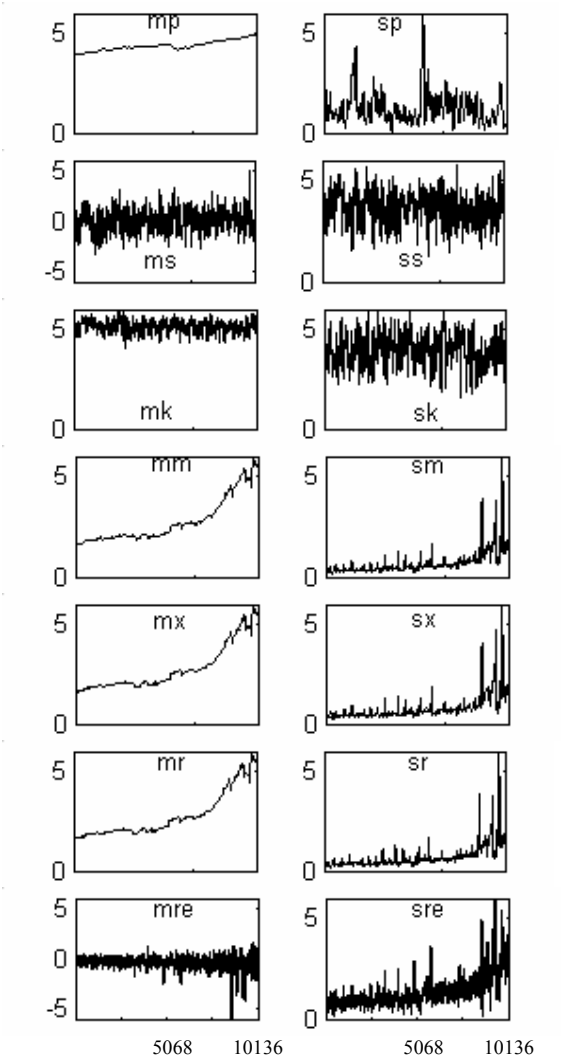
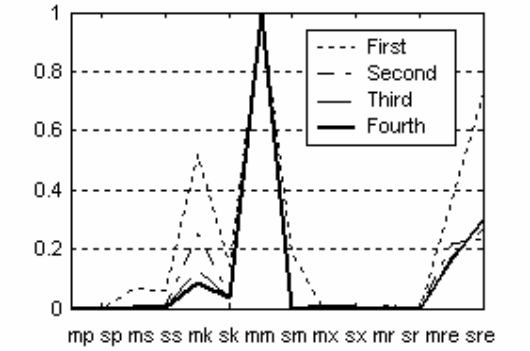


Figure 2. Features extracted from Test 1. (the Unit of the Horizontal Axes is "Second")

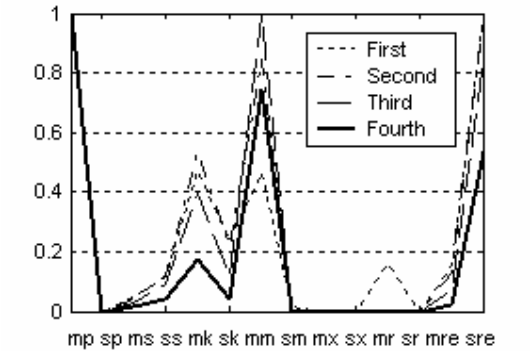
4. RESULTS

Fourteen hyperparameters are assigned to the fourteen feature candidates respectively. Following the steps (1) to (6), mentioned in Section 2, feature selection can be achieved. During the computation, the less relevant feature dimensions are effectively

suppressed as their controlling parameters are automatically reduced to zero or much smaller values than those of the relevant ones. Two examples are shown in the following figures.



(a)



(b)

Figure 3. Feature selection results of Test 1 and Test 4.

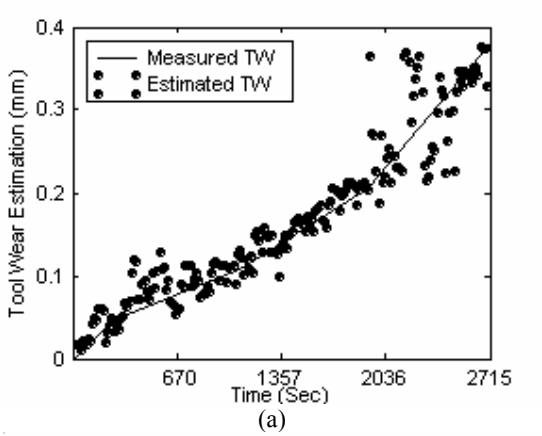
Figure 3 (a) shows the feature selection result of Test 1, in which the tool was gradually worn; (b) is the result of Test 4, in which micro-chipping phenomenon occurred. The vertical coordinates are the normalized magnitudes of the parameters assigned to candidate feature dimensions, where the solid bold line represents the result of the final results. Note that the power spectral density of the force signal is sensitive to Test 4, but not to Test 1. The feature selection results of all the experiments are listed in Table 3.

Test No.	Selected Feature Set
1	*{mm, mk, sk, mre, sre}*
2	*{mm, ss, mk, sk, mre, sre}*
3	*{mm, ss, mk, sk, mre, sre}*
4	*{mp, mm, ss, mk, sk, mre, sre}*
5	*{mp, mm, sk, mre, sre}*
6	*{mm, sk, mre, sre}*

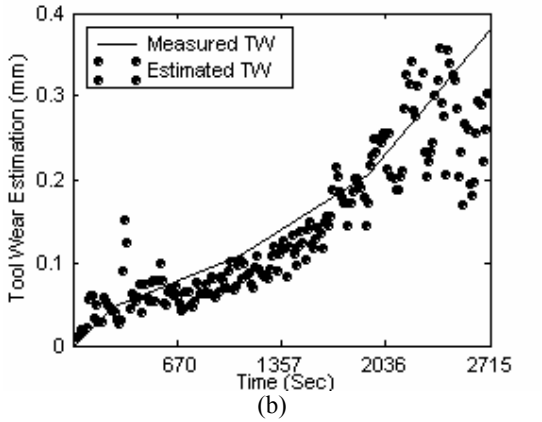
Table 3. Feature selection results of all the experiments.

The results can be summarized as follows. There are slight differences in the ARD feature selection manners between the gradually changing force signals and the dramatically changing ones. The differences mainly focus on whether the features related to the power spectral density are relevant or not. In order to make the selected feature set applicable for a wide range of conditions, we regard all of the features appearing in the table as relevant. Thus, we select *{mp, mm, ss, mk, sk, mre, sre}* as the relevant feature set.

In order to test the generalization capability of the SVR algorithm and to prove that the selected features are really more relevant than the rejected features, we did another experiment, whose conditions were different from all the previous experiments. The feature samples from Test 1 to Test 6 were used to train the regression network, while those of Test 7 were used to test it. The 7 selected features were first used in the training and testing process. Then the 7 rejected features were used to repeat the processes. The results are illustrated in the following figures.



(a)



(b)

Figure4. Comparison of tool wear estimation results.

Figure 4 (a) shows the results by using the selected features; (b) shows those by using the rejected ones. Define the averaged absolute error as the sum of all the absolute errors (differences between the measured wear values and the estimated wear values) divided by the

total number of estimated points. And define the accuracy as the ratio between the averaged absolute error and the final wear value. Then the accuracy of the first case is 5.6% with an error of 21.5 microns, and that of the second case is 9% with an error of 33.8 microns. It can be clearly seen from either the two figures or the two accuracies that the selected features are really more useful than the rejected ones.

However, even using the selected features the estimation results are still not very good. To improve the estimation performance, we proposed a moving average method. Specifically, we further processed the results given by the SVR algorithm ($f_{MP}(i)$) using the following equation

$$f_{MP}^{*}(i) = \sum_{j=i-mw+1}^{i} f_{MP}(j) \Big/ mw \quad \forall i \geq mw \qquad (25)$$

where *mw* is the size of the moving window, whose value is 4 in this paper. The idea behind this is to reduce the noise added to estimation values. As we assume a Gaussian distribution for the additive noise, whose form is given in (3) and (4), the estimation error should be centered at zero and distribute normally. This situation is illustrated in Figure 4 (a), where the estimation values oscillate up and down in the vicinities of the true values. Moving average methods could be a sufficient way to reduce the estimation error, because of the mutual cancellation among positive and negative errors. Figure 5 illustrates the results after moving average processing (the selected features were used). The accuracy in this case is 2.9% with an error of 11.1 microns, which is far better than the original results.

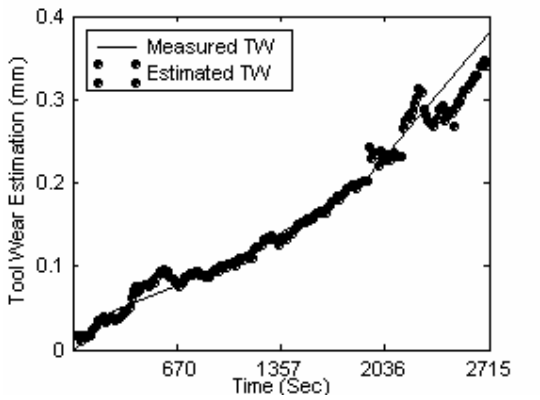


Figure 5. Results after moving average processing.

5. CONCLUSION

From the results obtained, it can be seen that embedding the ARD concepts into the SVR algorithm can effectively suppress the less relevant features by automatically reducing the magnitudes of their controlling parameters to zero or almost zero. Hence, even without any priori knowledge about the relevance of features, we can still get good estimation results through the ARD embedded SVR algorithm. Furthermore, as we continue using this method, we can accumulate sufficient knowledge about features that are more relevant in a specific application case. Consequently, we can discard the irrelevant features and just use the relevant ones for later applications. This can both enhance the regression performance and reduce the time needed for computation.

REFERENCE

[1]. Mohri, Bertok, and Sata, In process monitoring of tool breakage based on atuo-regressive model, IFAC, Session 3, measurement techniques, Maryland (1982).

[2]. Altintas, In-process detection of tool breakages using time series monitoring of cutting forces, Int. J. Mach Tools Manufact. Vol. 28, No. 2, 157-172, 1988.

[3]. Tarng, Hseih and Hwang, Sensing tool breakage in face milling with a neural network, Int. J. Mach Tools Manufact. Vol. 34, No. 3, 341-350, 1994.

[4]. Tansel and Mclaughlin, Detection of tool breakage in milling operations-II. the neural network approach, I. J. Mach Tools Manufact. Vol.33,No.4,545-558, 1993.

[5]. Kim and Choi, Development of a Tool Failure Detection System Using Multi-Sensors, Int. J. Mach. Tools Manufact., Vol. 36, No. 8, pp. 861-870, 1996.

[6]. Niu, Wong, Hong, An Intelligent Sensor System Approach for Reliable Tool Flank Wear Recognition, Int. J. Adv. Manuf. Technol., Vol. 14, 77-84, 1998.

[7]. Vapnik, The Nature of Statistical Learning Theory, New York: Springer-Verlag, 1995.

[8]. Tipping M.E., The relevance vector machine, Neural Information Processing Systems, 12, pp. 652–658. MIT Press, 1999.

[9]. W. L. Buntine, and A. S. Weigend, Bayesian back-propagation, Complex Systems, 5(6): 603–643, 1991.

[10]. David J. C. Mackay, Bayesian Methods for Neural Networks: Theory and Application, Lecture Notes for Neural Network Summer School.

[11]. Neal, R. M, Bayesian Learning for Neural Networks, Lecture Notes in Statistics, Springer, 1996.

[12]. C. K. I. Williams, Prediction with Gaussian processes: from linear regression to linear prediction and beyond, Learning and Inference in Graphical Models, 1998, Kluwer Academic Press.

[13]. W. Chu, S. S. Keerthi, and C. J. Ong, A unified loss function in Bayesian framework for support vector regression, In Proceeding of the 18th International Conference on Machine Learning, 2001, pp. 51–58.

[14]. D. J. C. MacKay, A practical Bayesian framework for back propagation networks, Neural Computation, 4(3), 1992, pp. 448–472.