Signal Stream Clustering for Tool-Revolution-Level Tool Condition Monitoring in Milling Process

Si Jie Phua [1], Xiang Li [1], Wee Keong Ng [2],
Beng Siong Lim [1], Weixiang Zhong [1], and Junhong Zhou [1]

[1] *Singapore Institute of Manufacturing Technology, 71, Nanyang Drive, Singapore 638075*
{sjphua,xli,gbslim,wxzhong,jzhou}@SIMTech.a-star.edu.sg
[2] *Nanyang Technology University, 50 Nanyang Avenue, Singapore 639798*
wkn@acm.org

ABSTRACT

The researches in tool condition monitoring often collect large amount of sensor signal data from experiments to study the complex tool condition relationships with signals. In order to provide new light into this process on a real-time basis, it is critical to identify and detect abnormality at the lowest resolution possible so that the wear behavior on each flute within a tool revolution can be clearly shown. A signal stream clustering method is developed to separate numerous tool-revolution signals into similar groups, each representing a specific set of corresponding events. In our experiment, the 1000 tool-revolution signals in force signal stream are grouped into 5 clusters. These clusters in turn provide a visual mean to assess the tool condition at the most detailed level. In addition, the clusters also enable complex tool condition relationships to be established from the signatures of each set of events.

1 INTRODUCTION

Tool condition monitoring in milling process is challenging because of the intermittent contacts between workpiece and the edges of cutting tool. The intermittent contacts induce large frequency components that disable direct monitoring with single reading from sensors (Rehorn *et al.*, 2005). To monitor wear state of cutting tool, existing researches collect large amount of sensor data and establish various feature extraction and modeling methodologies to develop wear state prediction model (Tansel *et al.*, 2005; Chung and Geddam, 2003; Amer *et al.*, 2007; Hong *et al.*, 2006; Zhu *et al.*, 2008b; Li *et al.*, 2006; Aliustaoglu *et al.*, 2009). Sets of extracted features are to be selected according to the target conditions (Zhu *et al.*, 2008a). In addition, the prediction model requires threshold adjustment to strike a balance between sensitivity and accuracy of predicting the tool damage whenever it is used on different conditions (Amer *et al.*, 2007).

To provide more insights on the tool condition with collected signal in real time basis, we propose to monitor at the lowest possible resolution. As cutting tool cuts the workpiece revolution by revolution, we believe that the visualization of signal profile at tool revolution level closely reflects the tool conditions. For example, Figure 1 shows that the force profiles can identify the number of engaged cutting edges. As far as we know, this issue has not been addressed in existing research.

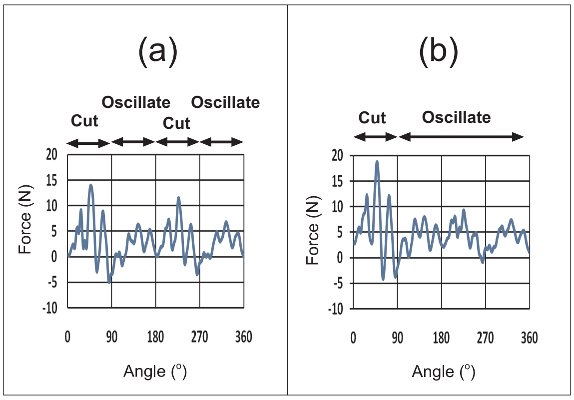

Figure 1: (a) This tool-revolution force profile of end-milling shows that two cutting edges are cutting. For a two-flute cutting tool, this profile is desired as both cutting edges share the workload. (b) This tool-revolution force profile of end-milling shows that only one cutting edge is cutting. For a two-flute cutting tool, this profile indicates imbalance setup or worn of one flute that causes poor milling quality and fast tool wear rate.

However, the uncountable revolutions in high-data-rate signal stream create barrier for machine operators to visualize and characterize the milling performance at the tool revolution level. At the early stage of this work, we discovered that most tool-revolution signals are alike, which prompted us that representing similar individual signals with clusters would reduce the data size. Furthermore, with the clustering, the obtained major clusters and evolution trends of tool-revolution signals could enable visualization to assess both milling performance and its stability.

The clustering at tool revolution level is composed of two steps. The first step is to segment the signal data into tool revolution level while the second step is to cluster the tool-revolution signals into visualizable information. As signal data comes very fast, these two steps have to be performed in real time. In this paper, we elaborate more on the clustering in real time which is the challenging task.

In our clustering technique, retaining the signal shape is crucial as the signal shape is the basis for machine operator to characterize the milling performance. To retain the signal shape, we adopt a two-phase scheme (Aggarwal *et al.*, 2003) for our clustering technique. In the two-phase scheme, the closely-similar tool-revolution signals from raw stream data are firstly grouped into micro-cluster in online component. Then, macro-level clustering is performed on the micro-cluster in offline component when visualization is requested. We evaluate our clustering technique on force signal data during end-milling process by comparing the visualization of obtained clustering information against the raw signal. We aim to show that the obtained clustering information of force signal provides instant view on major behaviors and stability of milling process. The visualization enables machine operators to characterize the milling performance in-situ and hence reduce the milling process setup time.

2 RELATED WORKS

2.1 Tool Condition Monitoring

The researches in tool condition monitoring aim to provide in-situ monitoring with indirect sensors, such as force and acoustic emission sensors (Rehorn *et al.*, 2005). In (Chung and Geddam, 2003), the force, torque and acoustic emission signals that are previously used to monitor tool condition of turning and drilling are evaluated for prediction of the tool wear state in end-milling process. The result shows that the prediction of tool wear state is feasible with the extracted features from torque and acoustic emission signals.

To improve the prediction accuracy of tool wear state, Hong *et al.* (2006) highlights that the data selection is important during training of prediction model. The proposed data selection strategy is validated to provide effective and efficient training set. The training set allows fast and reliable model training on Support Vector Machine to deliver accurate tool wear prediction model. As prediction accuracy is affected by noise component in micro-milling, blind source separation technique is deployed for signal preprocessing before feature extraction and model training from collected force signal (Zhu *et al.*, 2008b). With the proposed signal preprocessing technique, the localized model developed with Hidden Markov Models is validated to be noise-robust.

In (Zhu *et al.*, 2008a), the prediction accuracy is improved by selecting critical features that are sensitive to tool wear state. To perform feature selection, Fisher's Linear Discriminant Analysis is modified to select the critical features. The selected feature set is shown to be superior to that by other methods for training of Hidden Markov Model. Sun *et al.* (2008) further demonstrates the effectiveness of feature selection by conducting a case study in titanium machining. Automatic Relevance Detection is used to select effective feature set for training on Support Vector Machine.

To incorporate aforementioned steps that are used to derive accurate tool condition monitoring model, these steps are mapped with those in generic model development (Li *et al.*, 2006). The mapped steps are demonstrated by case study to predict remaining useful lifetime of cutting tools. To incorporate the prediction capability of various sensors, a two-stage fuzzy logic scheme is proposed (Aliustaoglu *et al.*, 2009). The first stage creates multiple fuzzy inference models for each sensor while the second stage delivers sensor fusion model based on responses from the primary fuzzy inference models.

Amer *et al.* (2007) points out that the conventional models for tool condition monitoring requires retraining or threshold adjustment whenever it encounters new operation. The retraining or threshold adjustment is usually an exhaustive effort. To minimize the effort, Amer *et al.* (2007) incorporates technique that can quickly adjust the focus frequency band for different application. The signal at focus frequency band is further processed to derive health index of the cutting tools. With the similar objective, genetic algorithm is deployed to automate the threshold adjustment on existing analytical force model (Tansel *et al.*, 2005). The proposed methodology is able to estimate machining parameters and provide indicators for tool damage.

In (Rehorn *et al.*, 2005), the tool condition monitoring researches in turning, drilling and milling is reviewed. The conclusion states that the simpler a tool condition monitoring system is, the less likely it is to fail. Thus, we propose to provide milling performance characterization at tool revolution level that is the fundamental level of signal. As far as we know, providing the fundamental view with collected signal is yet to be investigated.

2.2 Time Series Stream Clustering

The signal stream to be clustered in our works can be seen as time series data. A time series is a sequence of data points that are measured at succesive time. To cluster the time series stream in real time, Piecewise Aggregate Approximation (PAA) technique is proposed to transform the real-value time series into symbolic representation (Lin *et al.*, 2003). The transformation reduces the dimensionality of time series. To further reduce the size of time series data, the data is clipped using median of original series into binary series (Bagnall and Janacek, 2005). The experiment results show that the clustering speed increases without losing accuracy. To further enhance the speed of clustering times series, the real valued time series data are clipped into binary series before clustering by incorporating PAA and bi-clipped processes (Li *et al.*, 2007).

In (Rodrigues *et al.*, 2008), a hierarchical divisive clustering structure is maintained. The hierarchical structure is updated upon receiving a fixed number of incoming time series data. The experiment results suggest that the proposed algorithm adapt to changes in time series.

To perform meaningful time series clustering (Keogh and Lin, 2005), we aim to retain the signal shape as detailed as possible. To achieve the shape retaining while clustering in real time, we adopt the two-phase clustering scheme proposed in (Aggarwal *et al.*, 2003). In (Aggarwal *et al.*, 2003), the stream data is firstly clustered into micro-clusters upon arrival during online updating phase. Snapshots of micro-clusters are kept according to pyramidal time frame. During offline clustering phase, the offline clustering is performed on micro-clusters at requested snapshots with modified k-means algorithm.

3 METHODOLOGY

To characterize milling performance with collected signal stream at tool revolution levels, we firstly segment the signal stream into tool-revolution signals in real time. Then, these signals are grouped into micro-clusters in real time by our clustering technique. When characterization is requested, macro-clustering is performed on micro-clusters to provide better visualization.

3.1 Signal Stream Segmentation at Tool Revolution Level

The signal stream segmentation is obtained with the calculated data length of one tool revolution. The data length of one tool revolution, L can be calculated with the sampling frequency and spindle speed that are known a priori. Assume sampling frequency is N Hz and the spindle speed is S RPM (revolutions per minute), the data length of one tool revolution, L can be calculated with Equation 1:

$$L = \frac{N \cdot 60}{S} \tag{1}$$

Hence, the tool-revolution signals are obtained by taking every L consecutive data points as a tool revolution. Figure 2 compares the visualization of raw signal stream with that after segmentation. From Figure 2(b), it can be seen that both edges of cutting tool are cutting during the four tool revolutions. As uncountable tool revolutions are segmented from signal stream, checking these revolutions one by one is time consuming if not impossible. Thus, clustering on these segmented signals provides faster visualization on milling performance.

3.2 Clustering on Tool-Revolution Signals

Our clustering algorithm aims to group numerous segments of tool-revolution signals into countable clusters. The R segments of tool-revolution signals are grouped into k clusters where $k \ll R$. In each cluster, the following data is maintained for retaining the signal shape at tool revolution level.

- Upper bound u_i of all the tool-revolution signals belonged to the cluster at time instance t_i, $1 \leq i \leq L$
- Lower bound l_i of all the tool-revolution signals belonged to the cluster at time instance t_i, $1 \leq i \leq L$
- Average a_i of all the tool-revolution signals belonged to the cluster at time instance t_i, $1 \leq i \leq L$

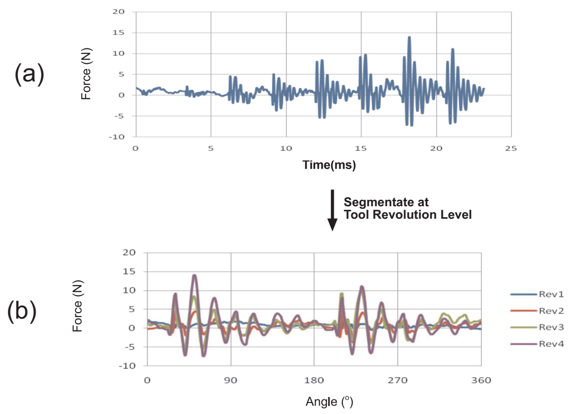

Figure 2: (a) Four tool revolutions of force signal collected during entrance period of cutting tool. (b) Four signal segments at tool revolution level indicate both cutting edges are cutting during entrance period.

- Number n of the tool-revolution signals belonged to the cluster

The segments are grouped into clusters according to distance metric in Equation 2. Suppose c_1 and c_2 are different clusters, the distance between them can be computed with following equation:

$$ClusDist(c_1, c_2) = \max_i(PtDist(i)) \tag{2}$$

where

$$PtDist(i) = \max(c_1.u_i - c_2.l_i, c_2.u_i - c_1.l_i). \tag{3}$$

To compute the distance between two segments, the upper bound u_i and lower bound l_i can be replaced by the signal value of the segments at time instance t_i. The smaller the distance between two segments, the higher the priority for two segments to be clustered together.

To cluster the numerous segments in real time, we adopt a two-phase scheme (Aggarwal *et al.*, 2003) for our clustering technique. In the two-phase scheme, the closely-similar tool-revolution signals from raw stream data are firstly grouped into micro-cluster in online component. Then, macro-level clustering is performed on the micro-cluster in offline component when visualization is requested. Both clustering processes utilize the HIECLUS algorithm introduced in Algorithm 1.

The HIECLUS algorithm is developed on the concept of hierarchical clustering. It is the incremental component to allow data stream clustering. By performing HIECLUS algorithm, the R cluster instances is grouped into k cluster instances where $k \leq R$. The algorithm firstly computes distance metric for all the clusters at line 2. At line 3 and 4, the algorithm searches for candidate clusters to be combined. Two clusters with the lowest distance are combined. During combination, the algorithm updates the cluster information for the combined cluster at line 5 to 17. The

Algorithm 1 HIECLUS$(c, thre, trend)$

Require: Vector c contains $R \geq 2$ cluster instances, $thre \geq 0, 1 \leq trend_i \leq c.size() \;\forall 1 \leq i \leq T$
Ensure: Vector c contains $1 \leq k \leq R$ cluster instances, $1 \leq trend_i \leq c.size() \;\forall 1 \leq i \leq T$

1: Declare d as a 2 dimensional vector
2: $d(i,j) \leftarrow ClusDist(c_i, c_j) \;\forall 1 \leq i < j \leq R$
3: **while** $\min d(i,j) \leq thre$ **do**
4:     $[iMin, jMin] \leftarrow \operatorname{argmin}_{i,j} d(i,j)$
5:     $newn \leftarrow c_{iMin}.n + c_{jMin}.n$
6:     **for** $k = 1$ to L **do**
7:         **if** $c_{iMin}.u_k < c_{jMin}.u_k$ **then**
8:             $c_{iMin}.u_k \leftarrow c_{jMin}.u_k$
9:         **end if**
10:         **if** $c_{iMin}.l_k > c_{jMin}.l_k$ **then**
11:             $c_{iMin}.l_k \leftarrow c_{jMin}.l_k$
12:         **end if**
13:         $isum = c_{iMin}.a_k \cdot c_{iMin}.n$
14:         $jsum = c_{jMin}.a_k \cdot c_{jMin}.n$
15:         $c_{iMin}.a_k \leftarrow \frac{isum+jsum}{newn}$
16:     **end for**
17:     $c_{iMin}.n \leftarrow newn$
18:     Remove c_{jMin} from vector c
19:     Remove $d(i,j)$ from vector $d \;\forall j = jMin$
20:     **for** $k = 1$ to T **do**
21:         **if** $trend_i = jMin$ **then**
22:             $trend_i \leftarrow iMin$
23:         **end if**
24:         **if** $trend_i > jMin$ **then**
25:             $trend_i \leftarrow trend_i - 1$
26:         **end if**
27:     **end for**
28: **end while**
29:
30: **return** $[c, trend]$

search and combination of two clusters are performed until the minimum distance between every two clusters is larger than $thre$. After combination, the trend of cluster evolution along the timeline is updated at line 20 to 27.

In order to get representative clusters, retaining the signal shape is a crucial requirement for the clustering algorithm. The shape retaining is achieved by updating upper and lower bounds when signal segments or clusters are combined that are shown at line 7 to 12 of Algorithm 1.

The HIECLUS algorithm is performed in the online clustering component ONCLUS whenever a tool-revolution signal is received. The tool-revolution signal is obtained from methodology introduced in Subsection 3.1. Each tool-revolution signal is assigned to a micro-cluster ID with the online clustering algorithm introduced in Algorithm 2. The evolution of signal is tracked with the vector $trend$ that consists of a series of micro-cluster ID along the timeline. Hence, the raw signal can be represented with the micro-clusters and evolution trend that occupy much smaller memory space. In addition, the clustering threshold is adapted to the range of signals with preset $ratio$. The $ratio$ acts like an adaptive threshold for the clustering. The larger the $ratio$, the smaller the number of resultant micro-clusters and hence the smaller memory space occupied.

Algorithm 2 ONCLUS$(ratio)$

Require: $0 \leq ratio \leq 1$

Declare c as 1 dimensional vector of cluster instances
Declare $trend$ as 1 dimensional vector of integers
$max \leftarrow 0$
$min \leftarrow 0$
while data stream is active **do**
    read tool-revolution record r
    $i \leftarrow c.size() + 1$
    $c_i.u_k \leftarrow r_k \;\forall 1 \leq k \leq L$
    $c_i.l_k \leftarrow r_k \;\forall 1 \leq k \leq L$
    $rmax \leftarrow \max r_k$
    $rmin \leftarrow \min r_k$
    if $max < rmax$ **then**
        $max \leftarrow rmax$
    end if
    if $min > rmin$ **then**
        $min \leftarrow rmin$
    end if
    $t \leftarrow (max - min) \cdot ratio$
    $j \leftarrow trend.size() + 1$
    $trend_j \leftarrow i$
    $[c, trend] \leftarrow$ HIECLUS$(c, t, trend)$
end while

Algorithm 3 OFFCLUS$(ratio)$

Require: $0 \leq ratio \leq 1$

Declare tc, mc as 1 dimensional vector of cluster instances
Declare mtr as 1 dimensional vector of integers
$tc \leftarrow c$ in ONCLUS
$max \leftarrow \max tc_i.u_k \;\forall 1 \leq i \leq tc.size(), 1 \leq k \leq L$
$min \leftarrow \max tc_i.l_k \;\forall 1 \leq i \leq tc.size(), 1 \leq k \leq L$
$t \leftarrow (max - min) \cdot ratio$
$mc_i.u_k \leftarrow tc_i.a_k \;\forall 1 \leq i \leq mc.size(), 1 \leq k \leq L$
$mc_i.l_k \leftarrow tc_i.a_k \;\forall 1 \leq i \leq mc.size(), 1 \leq k \leq L$
$mtr \leftarrow trend$ in ONCLUS
$[mc, mtr] \leftarrow$ HIECLUS(mc, t, mtr)

To further reduce the data size for simplicity and clarity of visualization, the micro-clusters generated by ONCLUS is further grouped into macro-cluster by offline clustering component OFFCLUS. The OFFCLUS in Algorithm 3 perform reclustering with the average signals of the clusters. The clustering threshold is also adapted to the range of signals with preset $ratio$. The objective is to group as many similar signals as possible together while retaining the signal shape.

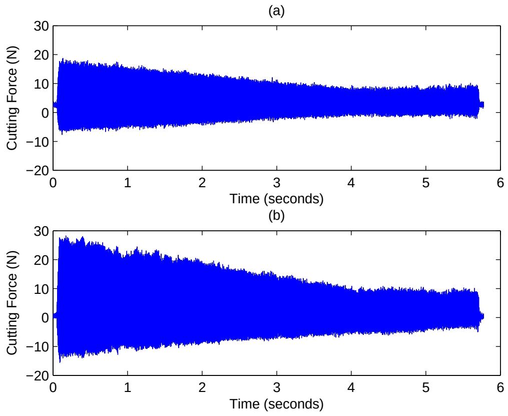

Figure 3: (a) Force signal of one cutting line in cutting direction when cutting tool is new. (b) Force signal of one cutting line in cutting direction when cutting tool degrades. Plotting the raw force signal shows the difference in trend but does not show the difference at tool revolution level.

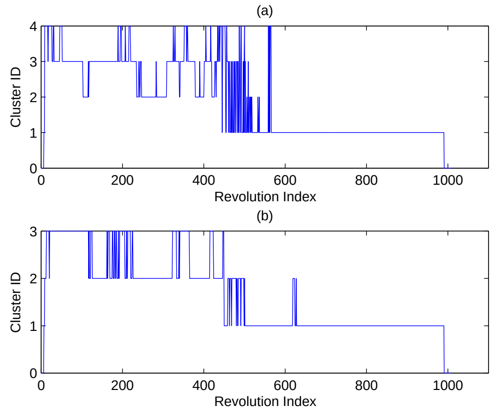

Figure 4: (a) Cluster evolution trend of force signal in Figure 3a where cutting tool is new. (b) Cluster evolution trend of force signal in Figure 3b where cutting tool degrades. The cluster ID is sorted according to the range of the signals it represents. Thus, the decreasing trends of signal magnitude that can be observed in Figure 3 are also reflected here.

4 RESULT AND DISCUSSION

To show the effectiveness of our clustering technique, we conduct our analysis on force signal. The force signal data is collected with sampling rate of 50kHz while a 2-flute end milling cutting tool is used to cut titanium. The difference on force signal between new tool and degraded tools is shown with our clustering technique. The portions of force signal used for analysis are plotted in Figure 3.

4.1 Tool Condition Visualization with Cluster Analysis

By applying our clustering technique on the signals in Figure 3, we provide more detailed information on condition and performance of the 2-flute end milling cutting tool with Figure 4 and 5.

Figure 4 compares the cluster evolution trends of the signals when cutting tool is new with that when cutting tool degrades. To get the trend, the signals in Figure 3 are segmentated at tool revolution level as mentioned in Subsection 3.1. Then, the segmented signals are clustered by online and offline clustering components. The adaptive clustering threshold $ratio$ is set to 0.3 in this case. As the result of clustering, each segmented signal is assigned with a cluster ID. The series of cluster ID forms the cluster evolution trend. From Figure 4, both of the trends are similar. This observation also accords with Figure 3. In addition, sudden shift across non-neighbor clusters is rare. Thus, the milling process is stable from the signal point of view.

Figure 5 compares the main clusters of the signals when cutting tool is new with that when cutting tool degrades. The signal magnitude in clusters of new tool is smaller than that of degraded tool. In addition, Figure 5a shows that the new cutting tool cuts with both cutting edges while Figure 5b shows that the degraded cutting tool has imbalance cutting with its cutting edges. This tool-revolution observation is not apparent in Figure 3.

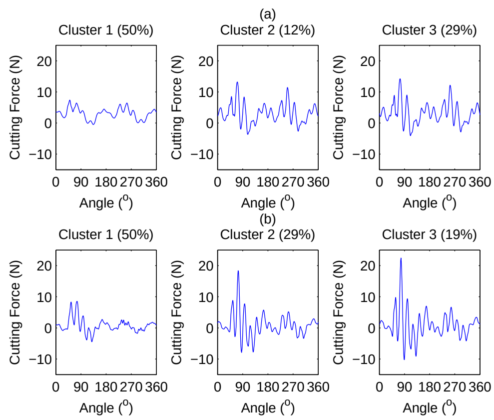

Figure 5: (a) Two groups of peaks appear on all the main tool-revolution clusters along the cutting line when cutting tool is new. The two peak groups show both cutting edges of the cutting tool are cutting. (b) Only one obvious group of peaks appear on all the main tool-revolution clusters along the cutting line when cutting tool degrades. It shows that one of the cutting edges plays more important role than the other cutting edges.

Figure 4 and Figure 5 shows that our clustering technique provides both the overall and tool-revolution views on force signals collected. With our clustering techniques, the 1010 tool-revolution signals are compressed into 5 clusters in Figure 4a. Each cluster occupies 3 times more memory spaces compared with a tool-revolution signal as indicated in Subsection 3.2. Thus, the compression ratio of force signals by our clustering algorithm is about $\frac{1010}{5 \cdot 3} = 67$ times.

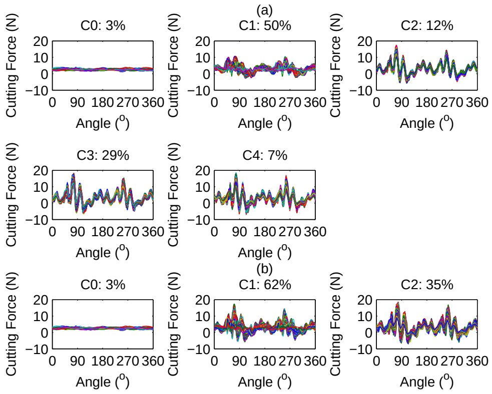

Figure 6: (a) The clusters are generated with $ratio$ set to 0.3. All the signals in the same cluster look similar. (b) The clusters are generated with $ratio$ set to 0.4. Many different signals are grouped into cluster C1.

4.2 Coherence of Tool-Revolution Cluster

From Figure 6, the effect on different setting of adaptive clustering threshold $ratio$ is shown. Larger $ratio$ results in smaller number of clusters and larger compression of information. However, larger $ratio$ also reduces the accuracy to represent the individual signals by clusters.

5 CONCLUSION

In this paper, we have studied the problem of monitoring tool condition in real time. We have argued that tool revolution, which is the lowest resolution, retains detailed tool condition information and hence the need of monitoring in tool-revolution view. We have addressed the research challenge to provide visualization in both tool-revolution and overall views in real time with our two-phase clustering technique.

Our clustering technique have been implemented and tested on force signal generated by two-flute cutting tool. The visualization generated with our clustering technique is effective in comparing the difference of tool condition at tool revolution level and showing the stability of milling process in real time. Thus, our clustering technique enables machine operators to access the tool condition and performance in the most detailed level.

For future work, we intend to obtain failure signatures and provide diagnosis by correlating the obtained clusters with failure events in real time.

ACKNOWLEDGMENTS

The authors wish to thanks Alignment Tool(S) Pte Ltd's management team and Mr M.H. Seet, Q. Xing together with all the operational staff for their practical advice, commitments and valuable contributions in the experiment setup and testing. The contributions and expertises from other members of the SIMTech team, particularly K.C. Shaw and S. Huang are greatly appreciated.

REFERENCES

(Aggarwal *et al.*, 2003) Charu C. Aggarwal, T. J. Watson, Resch Ctr, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for clustering evolving data streams. In *VLDB*, pages 81–92, 2003.

(Aliustaoglu *et al.*, 2009) C. Aliustaoglu, H. M. Ertunc, and H. Ocak. Tool wear condition monitoring using a sensor fusion model based on fuzzy inference system. *Mechanical Systems and Signal Processing*, 23(2):539–546, 2009.

(Amer *et al.*, 2007) W. Amer, R. Grosvenor, and P. Prickett. Machine tool condition monitoring using sweeping filter techniques. *Proceedings of the Institution of Mechanical Engineers, Part I (Journal of Systems and Control Engineering)*, 221(I1):103–117, 2007.

(Bagnall and Janacek, 2005) A. Bagnall and G. Janacek. Clustering time series with clipped data. *Machine Learning*, 58(2-3):151–178, 2005.

(Chung and Geddam, 2003) K. T. Chung and A. Geddam. A multi-sensor approach to the monitoring of end milling operations. *Journal of Materials Processing Technology*, 139:15–20, 2003.

(Hong *et al.*, 2006) G. S. Hong, J. Sun, Y. S. Wong, M. Rahman, and Z. G. Wang. Effective training data selection in tool condition monitoring system. *International Journal of Machine Tools and Manufacture*, 46(2):218–224, 2006.

(Keogh and Lin, 2005) E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8(2):154–177, 2005.

(Li *et al.*, 2006) Xiang Li, Junhong Zhou, Hao Zeng, Yoke San Wong, and Geok Soon Hong. An intelligent predictive engine for milling machine prognostic monitoring. *2006 4th IEEE International Conference on Industrial Informatics*, pages 1075–1080, 2006.

(Li *et al.*, 2007) Weimin Li, Liangxu Liu, and Jiajin Le. Clustering streaming time series using cbc. In *ICCS '07: Proceedings of the 7th international conference on Computational Science, Part III*, pages 629–636, Berlin, Heidelberg, 2007. Springer-Verlag.

(Lin *et al.*, 2003) Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *In Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 2–11. ACM Press, 2003.

(Rehorn *et al.*, 2005) A.G. Rehorn, Jin Jiang, and P.E. Orban. State-of-the-art methods and results in tool condition monitoring: a review. *International Journal of Advanced Manufacturing Technology*, 26(7-8):693–710, 2005.

(Rodrigues *et al.*, 2008) Pedro P. Rodrigues, Joao Gama, and Joao Pedroso. Hierarchical clustering of time-series data streams. *Knowledge and Data Engineering, IEEE Transactions*, 20(5):615–627, 2008.

(Sun *et al.*, 2008) Jie Sun, Yoke San Wong, Hong Geok Soon, Mustafizur Rahman, and Zhigang Wang. Identification of feature set for effective tool condition monitoring - a case study in titanium machining. *4th IEEE Conference on Automation Science and Engineering, CASE 2008*, pages 273–278, 2008.

(Tansel *et al.*, 2005) I. N. Tansel, W. Y. Bao, N. S. Reen, and C. V. Kropas-Hughes. Genetic tool monitor (gtm) for micro-end-milling operations. *International Journal of Machine Tools and Manufacture*, 45(3):293–299, 2005.

(Zhu *et al.*, 2008a) K. P. Zhu, G. S. Hong, and Y. S. Wong. A comparative study of feature selection for hidden markov model-based micro-milling tool wear monitoring. *Machining Science and Technology*, 12(3):348–369, 2008.

(Zhu *et al.*, 2008b) K. P. Zhu, Y. S. Wong, and G. S. Hong. Noise-robust tool condition monitoring in micro-milling with hidden markov models. In *Soft Computing Applications in Industry*, pages 23–46. Springer, 2008.

Si-Jie Phua (Singapore Institute of Manufacturing Technology) is a research engineer at the Singapore Institue of Manufacturing Technology, Singapore. He received his BEng (1st Honors) from the School of Computer Engineering, Nanyang Technology University, Singapore, in 2006. He is currently a PhD candidate supervised by A/Prof Ng Wee-Keong in the School of Computer Engineering at Nanyang Technology University. His current research interests are data stream mining and sensor signal analysis.

Xiang Li (Singapore Institute of Manufacturing Technology) is a research scientist at the Singapore Institue of Manufacturing Technology, Singapore. She received her Ph.D from Nanyang Technological University, Singapore, in 2000. She has more than 15 years of experience in research and applications of data mining, artificial intelligence and statistical analysis, such as neural networks, fuzzy logic systems, data clustering and multiple regression modelling. She has led and was involved in many research and industrial projects, including computational intelligence for tool and surface integrity monitoring in precision machining, automould designs using intelligent optimisation and simulation for semiconductor equipment builder, automation and integration of manufacturing executive system (MES) for semiconductor in Taiwan, data warehousing for semiconductor in Singapore, rule-based AEC object library for building constructor, customer demand knowledge discovery in new product design for electronics in Singapore and web-based intelligent forecasting system. Her current research interest include data mining, machine learning, computational intelligence, decision support systems, intelligent forecasting, knowledge-based systems, real time equipment condition monitoring and performance prediction. She is a member of the Decision Sciences Institute, USA and member of IEEE.

Wee-Keong Ng (Nanyang Technological University) is an Associate Professor with the School of Computer Engineering at the Nanyang Technological University, Singapore. He received his PhD from the University of Michigan at Ann Arbor, USA, in 1996. He has published more than 200 refereed journal and conference articles in the area of data mining, web information systems, database and data warehousing and software agents. His journal papers have appeared in the ACM Transactions on Knowledge Discovery from Data, IEEE Transactions on Knowledge and Data Engineering, ACM Computing Surveys, IEEE/ACM Transactions on Computational Biology and Bioinformatics, Data and Knowledge Engineering Journal, World Wide Web Journal, and other major journals. He actively participates in international conference activities. He serves in the program committees of numerous international conferences annually and has held tutorials and talks at conferences and research seminars. His current research interests include data mining and analysis of large data sets, mining of time series data and sensor data, privacy-preserving data mining and information security. He is a member of IEEE, IEEE Computer Society and ACM.

Beng-Siong Lim (Singapore Institute of Manufacturing Technology) is a senior research scientist at the Singapore Institue of Manufacturing Technology, Singapore. He received PhD from the Production Engineering and Production Management, The University of Nottingham, England, in 1986. He led several teams to develop intelligent system for the characterisation of cutters in the green machining of aero-engine material and hardened steel; EDM electrode verification and offset insertion; flexible manufacturing systems; prosthetic knee joint implants; knowledge-based systems for daily production scheduling, process planning for progressive dies and modular fixture design; automated guided vehicles and robotics off-line programming. His current research interests include proactive performance engineering of manufacturing system; Development of reference models for the infusion of manufacturing knowledge into tools, process, equipment and systems; Multi-sensors fusion, performance, evaluation and application characterisation; Process and shop-floor monitoring, diagnostics and control; Equipment, plant and tools prognosis; EDM, tool engineering and process planning.

Weixiang Zhong (Singapore Institute of Manufacturing Technology) is a research engineer at the Singapore Institue of Manufacturing Technology, Singapore. He received his BEng from the School of Computer Engineering, Nanyang Technology University, Singapore, in 2009. His current research interest is data stream mining and sensor signal analysis.

Junhong Zhou (Singapore Institute of Manufacturing Technology) is a senior research engineer at the Singapore Institue of Manufacturing Technology, Singapore. She received her MEng from the Department of Electrical Engineering, National University of Singapore, Singapore. Her industrial experience includes intelligent monitoring system for grinding process, adaptive control, shop-floor Supervisory Control And Data Acquisition (SCADA) systems, enterprise resource management system, simulation study of factory layouts and optimisation of shop-floor utilisation. Her expertise and research interests include artificial intelligence, feature extraction and selection, fault diagnosis and prognosis, simulation, SCADA systems.