www.cadanda.com Computer-Aided Design AND APPLICATIONS™

A Neural Network Based Approach to 5-axis Tool-path Length Estimationfor Optimal Multi-cutter Selection

L. Geng[1], Y.F. Zhang[1*], and J. Y. H Fuh[1]

[1]National University of Singapore, mpezyf@nus.edu.sg

ABSTRACT

Compared to single-cutter machining, using multiple cutters in 5-axis finish machining of freeform surfaces can produce shorter tool-paths; hence the increased machining efficiency. In our previous work, a method to evaluate a cutter's accessibility at any point on a machining surface has been developed. In this paper, this method is used to identify feasible cutters and construct their machining regions. These cutters can make up many cutter combinations that can finish the entire machining surface, among which there will be an optimal set that produces the shortesttool-path. To find this optimal combination, we propose to use the tool of neural network to predict the tool-path length for a machining regionwithout actually generating the tool-path. The neural network is trained extensively with a large set of carefully designed training data extracted from actual machining jobs. Finally the validityof our method is proved with testing data sets that have never been exposed to the neural network before.

Keywords:five-axis machining; multi-cutter selection; neural network; tool-path length.
DOI: 10.3722/cadaps.2011.301-313

1 INTRODUCTION

In 5-axis machining, the cutter is given two more degrees of freedom, allowing it to be positioned in almost any direction relative to the machining surface. This feature not only enables the cutter to have better accessibility but also increases machining efficiency by giving the cutter a larger cutting profile [3]. However, this increased flexibility also brings the possibility of machining interference such as gouging and global collision, which inevitably complicates the process planning job. In this paper, we mainly deal with cutter selection, which is the first step in process planning.

Generally, large cutters have higher machining efficiency as they can remove more material in one path. However, at a point on a freeform surface, a cutter's interference-free posture range shrinks as its size gets larger. There will be a size limit beyond which the cutter will always produce machining interference at the point, no matter what posture it takes. On the contrary, smaller cutters tend to have large interference-free posture ranges, but smaller cutting profiles. This means for the same area, it will take smaller cutters more passes to finish, causing the machining efficiency to drop.

In most commercially available CAM (Computer-aided Manufacturing) systems, cutter selection is conducted manually based on the user's experience, which is both demanding and unreliable. On the other hand, there has been some reported work for automatic 5-axis cutter selection [5-6]. But the reported methods all have flaws in one or more aspects, such as incomprehensive interference checking or heavy computational load.

To meet the requirements for both quality and efficiency, one method is proposed in our previous work, where the largest cutter that can traverse the whole surface without causingany interference is selected for a machining job [8]. However, with this method, the selected cutteris constrained by the point with the most critical cutting condition. Suppose the machining surface has a small fraction of critical area, using a single small cutter for the rest of thenot-so-critical part of the surface will be a waste of the machine's capability. Therefore, we propose the idea of multi-cutter selection. The idea is to assign different cutters to different machining regions on the surface so that maximized machining efficiency can be achieved everywhere on the surface.

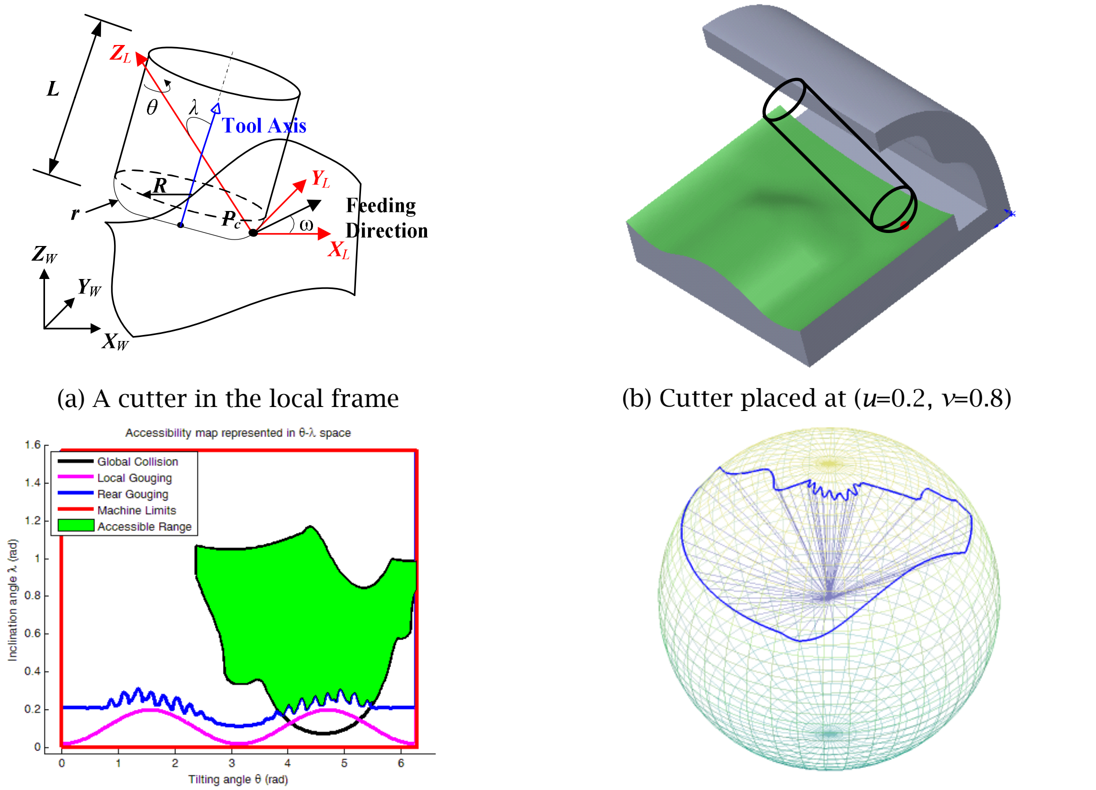
Currently, the idea of multi-cutter selection mainly focuseson 3-axis milling [1,4,11,12]. In this paper, we will extend the idea to 5-axis machining. First, accessibility checking is conducted for all available cutters, so that their accessible regions can be obtained. Based on this information, the candidate cutter sets that can finish the entire surface are identified. In each set, every cutter's actual machining regions will be allocated based on their respective accessible regions. Then a neural network is taken to predict the length of tool-path on each of these machining regions. The optimal cutter set will then be identified as the one that has the shortest total tool-path length. Finally, tool-path generation can be carried out with the optimal cutter set.

In section 2, a brief introduction for the concept and construction of the cutter accessibility map (A-map) will be presented. Section 3 presents algorithms for the identification of candidate cutter sets and the allocation of machining regions. In section 4, the construction and training of the neural network forpredicting tool-path length is carried out. Section 5 presents the testing results and section 6 gives the conclusion.

2 A-MAP: CONCEPT AND CONSTRUCTION

A part surface is described by a set of NURBS patches with C^2 continuity. A fillet-end cutter is described by its major radius*R*, minor radius r_f, and length*L*, as shown in Fig. 1(a). The *accessibility map* (A-map) is defined with respect to a cutter positioned at a point on the part surface. It refers to the posture range in terms of the two rotational angles, within which the cutter does not have any interference with the part and the surrounding objects. The two rotational angles are defined in the local frame at the point as shown in Fig. 1(a). The local frame (X_L, Y_L, Z_L) originates at the point of interest P_C with Z_L axis along the surface normal vector, X_L axis along the surface maximum principal direction, and Y_L axis the surface minimum principal direction. A cutter posture (λ, θ) means that the cutter's axis inclines λcounter-clockwiseabout Y_L axis and rotates θabout Z_L axis. Thus, the tool's axis vector can be given as $p = M_L^G(\sin(\lambda)\cos(\theta), \sin(\lambda)\sin(\theta), \cos(\lambda))^T$, where M_L^G is the transformation matrix between the local frame and the workpiece frame.

The algorithm used to determine the cutter's A-map at a point on the machining surface is called cutter accessibility (CA) algorithm. The CA algorithm finds the cutter's A-map at a point as the intersection of 4 accessible posture ranges, i.e., machine axis limits (ML), local-gouging-free region (LG), rear-gouging-free region (RG) and global-collision-free region (GC). Figs. 1(b-d) show an example of this process. The A-map for a cutter of size ($R = 5, r_f = 0.5, L = 60$)at point (*u*=0.2, *v*=0.8)as shown in Fig. 1(b) is constructed. The stock part contains a machining surface plotted in green and an arch-shape over-hang acting as non-machining obstacles. As shown in Fig. 1(c), the green region indicates the interference-free posture range. Finally, the interference-free posture range is mapped onto aunit-sphere in the local coordinate system as shown in Fig. 1(d). Any cutter posture within this range is considered interference free.



(a) A cutter in the local frame

(b) Cutter placed at (*u*=0.2, *v*=0.8)

(c) The A-map at (*u*=0.2, *v*=0.8)

(d) Accessible range mapped onto a unitsphere

Fig. 1: A-map and its construction.

The A-map at a point provides important knowledge for process planning. One direct application is in single cutter selection. To do this, the machining surface is first sampled into a point set with a certain density. Then the CA algorithm is applied at every sampled point for all the cutters in the cutter library. Finally, the optimal cutter will be selected as the largest cutter with non-empty A-maps at all the sampled points. For more information on A-map and its applications, please refer to [8].

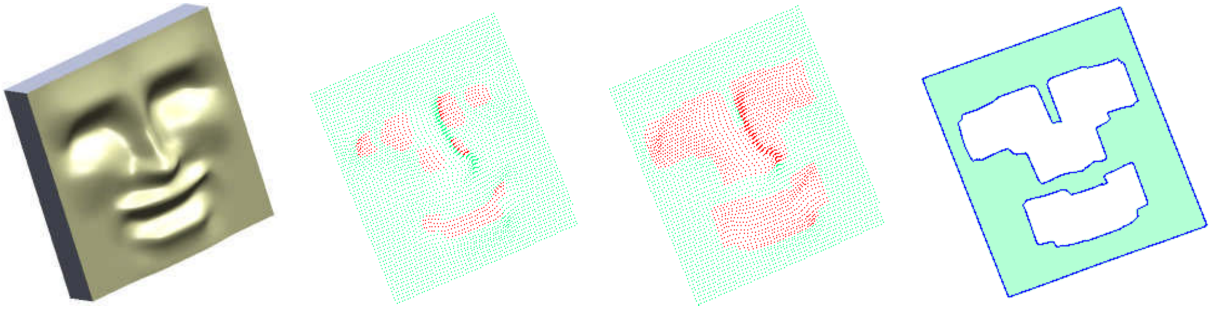
3 IDENTIFICATION OF CANDIDATE CUTTER COMBINATIONS

In a cutter library with*n*cutters, there could be a total number of $\sum_{i=1}^{n} C_n^i$ cutter combinations, most of which are null combinations that cannot finish the entire machining surface, or non-optimal combinations that will produce low machining efficiency. That is why we need to add some constraints to narrow down the range of search for optimal cutter combination. The input at this step would be: (1) The machining and non-machining surface provided in NURBS (2)A cutter library containing*n*cutters with their dimensions (R, r_f, L).

3.1 Evaluation of a Cutter's Accessible Area

The machining surface is firstly sampled into a point set of*m*×*m*points. A-map construction is carried out at all the*m*×*m*points for all the cutters in the cutter library. The cutters are then divided into three categories, i.e.,*accessible*cutters, which have non-empty A-maps at all the sampled points, *partially accessible* cutters, which have non-empty A-maps at some of the sampled points, and *inaccessible* cutters, which have non-empty A-maps at none of the sampled points. If a cutter library has more than one accessible cutter, all accessible cutters except the largest one are removed from the library to ensure maximized machining efficiency. Besides, the inaccessible cutters are also removed.

Fig. 2 shows an example ofthis cutter classification process fora partially accessible cutter on an example part, the “FACE” (see Fig. 2(a).). The surface is sampled into 201×201 points. The checking result is shown in Fig. 2(b) in which thered pointsmeans inaccessible (note that only half of the sampled points are plotted). Considering the fact that the gaps between the sampled points are not checked, we propose a refinement process to expand the inaccessible regionby a certain safety margin. For our example, the refined accessibility checking result is shown in Fig. 2(c).



(a) The “FACE”

(b) Original accessibility

(c) Accessibility after refinement

(d) Boundary extraction for accessible area

Fig. 2: An example of accessibility checking and accessible region identification.

Finally, the accessible areas for a cutter can be identified by findingthe boundaries of the inaccessible regions. To do that, we denote theinaccessible points by 0 and the accessible ones by 1to form a binary map of the surface. From there, a boundary tracing algorithm developed by Park and Choi[10] is employed. The procedure is applied and the result is plotted in Fig. 2(d). The accessible area is highlighted in green and the boundaries are in blue.

3.2 Finding Candidate Cutter Sets

A feasible cutter set should contain the only the largest accessible cutter left in the cutter library and some ofthe partially accessible cutters, so that the entire surface can be machined without any interference. Although forming all the feasible cutter sets is quite straightforward, the total number is very large. To reduce the overall search space for finding the optimal set, we propose to apply a practical constraint to shortlist the candidate cutter sets from the feasible sets.

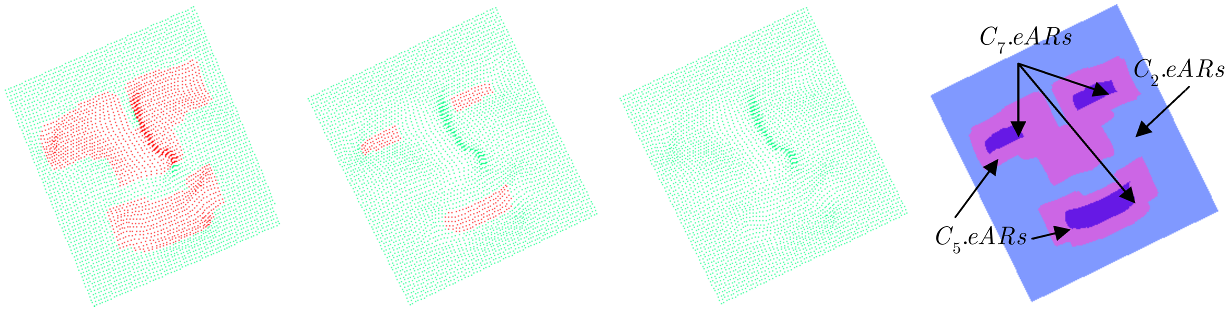
For a cutter in a feasible set, itsaccessible regions can be obtained as described in section 3.1. During tool-path generation, however, its actual machining regions will be equal or less than its accessible regions.This is due to the fact that the accessible regions of different cutters in the same set may overlap and in practice, the larger cutters should machine as much area as possible to maximize efficiency. Following this principle, we propose a method to assign machining regions to all the cutters in a feasible set from large to small. The regions that are assigned to a cutter are called the cutter’s effective accessible regions (*eARs*). Suppose we have a feasible set with *m*cutters listed from large to small as $CSet_i = \left\{C_{i1},\dots,C_{ij},\dots,C_{im}\right\}$. The *eARs* of the *j-th* cutter can be calculated recursively by:

$$C_{ij}.eARs = C_{ij}.ARs - C_{ij}.ARs \cap \left(\sum_{k=1}^{j-1} C_{ik}.eARs\right) \tag{1}$$

An example is given here to illustrate this procedure of obtaining the *eARs* of each cutter in a set. The “FACE” shown in Fig. 2 is used as the machining surface.The feasible cutter library contains 7 cutters listed from large to small in Tab. 1. The only accessible cutter is C_7 . A feasible set containing 3 cutters is constructedas $\left\{C_2, C_5, C_7\right\}$. The cutter’s accessible regions are shown in Figs. 3(a-c), respectively. Based on Eqn. (1), their *eARs* are calculated and plotted in different colors in Fig. 3(d). Proportionally, these 3*eARs* take up the machining area by52.7%, 37.2%,and 10.1%, respectively.

Cutter	*R (mm)*	r_f *(mm)*	*L (mm)*	*Cutter*	*R (mm)*	r_f *(mm)*	*L (mm)*
#1	10	0.5	60	#5	3	0.5	60
#2	8	0.5	60	#6	2.5	0.2	60
#3	6	0.5	60	#7	1.5	0.2	60
#4	4	0.5	60				

Tab. 1:The cutter library.



(a) $C_2.AR$ (b) $C_5.AR$ (c) $C_7.AR$ (d) *eARs* of the cutters

Fig. 3:An example for the construction of *eARs.*

Although large cutters can produce better material removal rate, the machining efficiency does not necessarily rise with the number of large cutters in the cutter set. According to [2], if a larger cutter has to travel a rather long distance to machine a very small region, the resultant time save may notbe worth the cost of unproductive air travel and extra cutter change. Based on this observation, it is necessary to restrict the number of cutters in each cutter set and make sure that each *eAR* is sufficiently large.For implementation, right after the *eARs* for a cutter are obtained, the proportion ratio of each *eAR* against the whole machining area is calculated. If the ratio of an *eAR* is less than a threshold value, say 10%, this *eAR* is deemed not feasible and removed.If all the *eARs* of a cutter are removed, the cutter itself will be removed. This heuristicapplies to all the cutters in a feasible set exceptthe accessible cutter. With this heuristic, the number of feasible cutter sets will drop greatly and the remaining sets are taken as the *candidate* cutter sets.

4 FINDING OPTIMAL CUTTER SET WITH NEURAL NETWORK

Assuming the same average feed rate and ignoring air-travel time, the cutter set with shortest overall tool-path length will produce the shortest machining time and therefore the highest machining efficiency. Thus, overall tool-path length is set as the criterion in our search for optimal cutter set. Generating tool-paths for all the candidate cutter sets would be too time-consuming.In our previous work [7], an attempt was made to rank the performance of cutter combinations using a criterion called Cutting Time Index (*CTI*), which is calculated as the machining surface area divided by the average strip width at all the sampled points. However, the method turned out to be less reliable than expected, probably due to the many assumptions made during the calculation of strip width.

Therefore, we here propose to predict tool-path length for a machining region using neural network without actually generating the tool-paths. The overall tool-path length will be calculated as the sum of tool-path lengths of all the machining regions.

4.1 Tool-path Generation Strategy

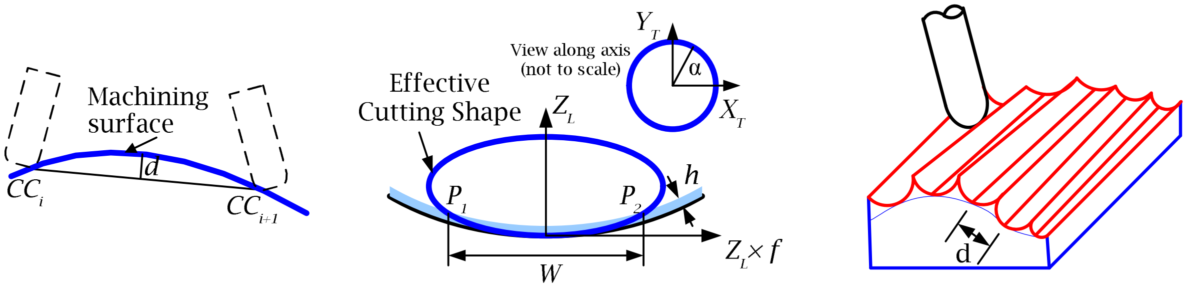
The tool-path pattern chosen in this study is iso-planar, also known as parallel cutting. The tool-paths compose of a series of cutter contact (CC) points, each with a cutter posture assigned to it. These CC points and cutter postures are generated based on certain strategies, which affect tool-path length greatly. In our study,the following strategies are employed:

(a) *The step-forward distances between neighboring CC points are maximized.* It is assumed that the cutter tip travels linearly between two neighboring CC points (see Fig. 4(a).). To maximize machining efficiency, the step-forward distance is chosen such that the deviation between the cutter tip trajectory and the machining surface reaches the given shape error tolerance.

(b) *The cutter posture is selected as the one that produces the maximum strip width.*The effective cutting shape of a cutter normal to the cutting directionis an ellipse (see Fig. 4(b).), which leaves a scallop on the machining surface. To maximize the scallop width, the two angles determining tool postures will be selected based on a heuristic:$\theta$is chosen such that the cutter is aligned with the cutting direction as much aspossible, λshould be kept as small as possible.

(c) *Side-step between neighboring paths is maximized.* The side-step value is selected so that the maximum scallop height between them stays just inside the allowable scallop height (see Fig. 4(c).).

Based on these strategies, a tool-path generation method is developed in our previous work. For a detailed explanation on how these strategies applied in tool-path generation, readers are advised to refer to [9].



(a) Step-forward calculation (b) Cutter posture selection at a CC (c) Side-step calculation

Fig. 4:The adopted tool-path generation strategies.

4.2 Predicting Tool-path Length with Neural Network

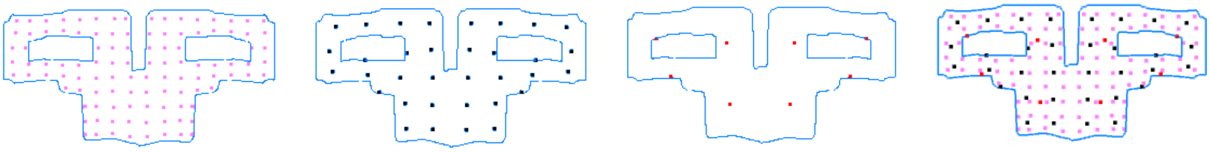
A neural network takes in a fixed number of inputs. For our study, these inputs shouldrepresent the geometric properties of the whole machining region faithfully. Besides, the machining strategies mentioned in section 4.1 should also be incorporated in the inputs, so that they can carry the necessary information about the machining characteristic over the whole machining region. Based on these two requirements, we propose to collect a fixed number of carefully distributed data points from a machining region and extract the inputs from these data points.

4.2.1 Input Data Extraction

As explained in Section 3, a high-density point set has already been sampled from the machining surface. The A-maps at these points have also been constructed. It will be quite convenient to selectthe input data points from the sampled points. The requirements forselection are: (1) fixed number (2) even distribution on the surface.

To meet these requirements, an iterative grid sampling algorithm is employed. Suppose a total number of*N*points are to be extracted. The bounding box of the region on the *x-y* plane is firstly segmented into*N*grids and all the sampled points in the region will fall into these grids.In each grid that has points inside, the centroid of the sampled points is calculated and the one nearest to the centroid is selected.

As the machining region may have an irregular shape, not all the grids will have sampled points inside. After one pass, the number of empty grids is recorded as N_e. Then sampling will be carried out again with these N_e grids. This process will be repeated till there is no empty grid left. An example is given in Fig. 5 showing this algorithm being applied to the first region of C_5 in Fig. 3. A total of 100 data points are extracted in three passes (see Figs. 5(a-c).). All the selected points are plotted in Fig. 5(d).



(a) First sampling (b) Second sampling (c) Third sampling (d) Selection results

Fig. 5: Data point sampling for neural network.

Since these data points are evenly distributed on the machining region, it is safe to say that they represent the geometric property well. Next, at each data point, we propose a machining characteristic parameter (*MCP*) that has direct relationship with the tool-path length.

The strip width determines the side-step between neighboring tool-paths, hence the number of tool-paths. It is determined by the cutter posture and the surface properties at the CC point. As given in strategy (b) in section 4.1, the cutter posture at a CC point is selected from the A-map with regard to the cutting direction. To maintain the consistency, the cutting direction taken to calculate the strip width at the extracteddata points is set asthe same one taken to generate the tool-paths for the region. With this cutting direction, the postures at the data points are selectedfrom their A-maps based on the samestrategy.

To calculate the strip width, the machining surface is first offset upward by *h*, which is the allowabletolerance for scallop height. On the plane normal to the cutting direction*f*, the cutter's effective cutting shape intersects this offset surface at two points, P_1 and P_2 (see Fig. 4(b)). The distance between them along direction $Z_L \times f$ is the strip width at the point, denoted by*W*. Suppose the cutter's posture is specified by the angle pair (λ, θ), on the plane normal to*f*, the cutter's effective cutting shape is given by:

$$E_L(\alpha) = \begin{pmatrix} X_L \\ Y_L \\ Z_L \end{pmatrix} = \begin{pmatrix} 0 \\ R\cos\lambda\sin\theta\cos\alpha + R\cos\theta\sin\alpha - R\cos\lambda\sin\theta \\ -R\sin\lambda\cos\alpha + R\sin\lambda \end{pmatrix} \tag{2}$$

where*α*is the angle that indicates a point's location on the cutting edge (see Fig. 4(b)).According to [13], the offset surface in the vicinity of the point can be approximated as:

$$Z_L = {}^{1}\!/_{2}\,\kappa_n Y_L^2 + h \tag{3}$$

where κ_n is the surface curvature at the data point on the plane normal to *f*. Eqns. (2) and (3)give the location of the intersection points. However, finding an explicit solution turns out to be difficult, which makes us turn to numerical methods. Suppose the location of the CC point on the cutter is indicatedby α_0. The two intersection points will fall on separate sides of the CC point, in the ranges $(\alpha_0 - Pi/2, \alpha_0)$ and $(\alpha_0, \alpha_0 + Pi/2)$ respectively. Their exact locations can be found based on Eqns. (2) and (3) using the simple bisection method. After this, the strip width W can be easily calculated.

In addition to the strip width, the area of the machining region, denoted by *A* is another factor that determines the length of the tool-path and thus must be considered in the *MCP*. The *MCP* of a data point is defined as *A/W*.

It is believed that with a set of data points of a certain density on the machining region, their *MCPs*will carry enough information to determine the tool-path length. Therefore, for our neural network, the inputs are taken as the *MCPs* at all the data points and the only output is specified as the tool-path length in a machining region. Tool-paths will be generated using the algorithm given in [9]. For implementation, we fix the number of data points, i.e., the number of inputs for the neural network to be 100. Finally, the data preparation algorithm for the neural network is presented as follows:

```
Algorithm: Data preparation for the neural network
Input: Cutter library; sampled points from the machining surface; cutting direction f.
Output:NN inputs for each machining region
Begin:
(1) Accessibility checking. Remove inaccessible cutters and redundant accessible cutters.
(2) Construct the accessible regions (ARs) for all the remaining cutters (section 3.1).
(3) Build the cutter sets. Make sure every set contains the accessible cutter.
(4)Foreach cutter inside a cutter set, construct the effective accessible regions (eARs). (Eqn. (1))
Surface area is recorded as $A_{surf}$ . Minimum area ratio for eARs is given as $\eta$ .
(5) In each cutter set, for every cutter except the accessible one:
    (a) Calculate the area for all the eARs. Remove the eARwhen $eAR / A_{surf} < \eta$ .
    (b) If for any cutter, $\{eARs\} = \varnothing$ remove the cutter set.
(6) Mark the remaining cutter sets as candidates. Record their eARsas machining regions.
(7) In each cutter set, for each machining region:
    (a) Select 100 data points using the grid sampling algorithm. (Section 4.2.1)
    (b) Calculate the MCPs at all the data points using f as the cutting direction. (Section 4.2.1)
(8) Generate tool-paths for each candidate set in direction f(Ref. [9]). Record actual tool-path length
in each region
(9) For candidate set i, the j-th machining region, a complete data set is given as:
$\left\{MCP_{ij}^{0} \ldots, MCP_{ij}^{100}\right\} \to TP_{ij}$ .
End
```

4.2.2 Training of the Neural Network

To generate the training data, 15 machining surfaceswere prepared. For each surface,several candidate cutter sets were selected. Each cutter set is coupled with a randomly selected cutting direction.One set of training data (100 inputs vs. 1 output) can be extracted from each machining region. For each cutter set, the machining surface will be partitioned into several machining regions. Thus we managed to collect 302 sets of training data from the 15 machining surfaces. Moreover, another 5 machining surfaces were created as testing samples. In the same way, 112 sets of testing data were generated from these surfaces. These testing data will not be exposed to the neural network during the training process.

The classic feed-forward back-propagation neural network (BPNN) has been chosen in this study. It is implemented with the MATLAB Neural Network Toolbox. The variance in our training sets is quite large. For a faster convergence, all the inputs (*MCPs*) have been normalized to the range of [-1, 1]. Only one hidden layer is used in our neural network. The number of training epochs is set as 20000, which as our training result shows, is enough for the networks to converge. In the toolbox, the error of the network is calculated as the mean absolute error (*MAE*). Suppose there are *m* sets of training data. The neural network's prediction of tool-path length is denoted as *TPN* and the real tool-path length is denoted by *TP*. *MAE* is calculated as:

$$MAE = \frac{\sum_{i=1}^{m} \left| TPN_i - TP_i \right|}{m} \tag{4}$$

During the training, the error goal is set as 0, which cannot be reached, so that the capabilities of the network can be exploited.

To determine the number of hidden neurons, a trial-and-error method was used. Neural networks with 3, 7, 13, 25 and 40 hidden neurons were constructed and trained. The best network is selected as the one that produces the minimum mean relative error *MRE*, which is calculated as:

$$MRE = \frac{1}{m}\sum_{i=1}^{m}\frac{\left|TPN_i - TP_i\right|}{TP_i} \tag{5}$$

Training results show that the neural network with 13 hidden neurons produces the smallest *MRE*. The neural network's structure is plotted in Fig. 6 and training parameters are given in Tab. 2.

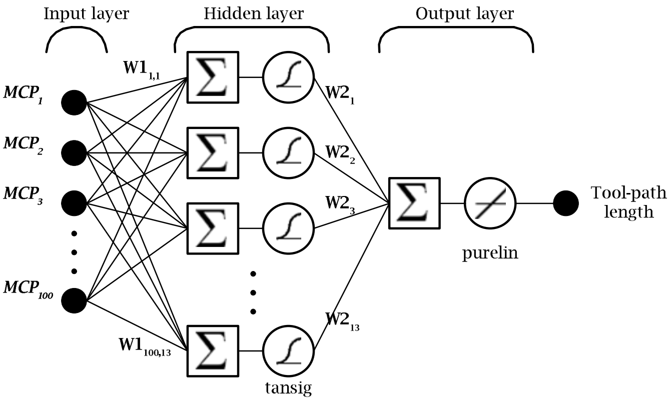


Fig. 6: Structure of the proposed neural network.

Parameters	Input neurons	hidden layer	hidden neurons	Learning rate	Training algorithm	Transfer function	Error goal	*Max epoch*
Values	100	1	13	0.05	ResilientBP	1st layer: tansig 2nd layer: linear	0	20000

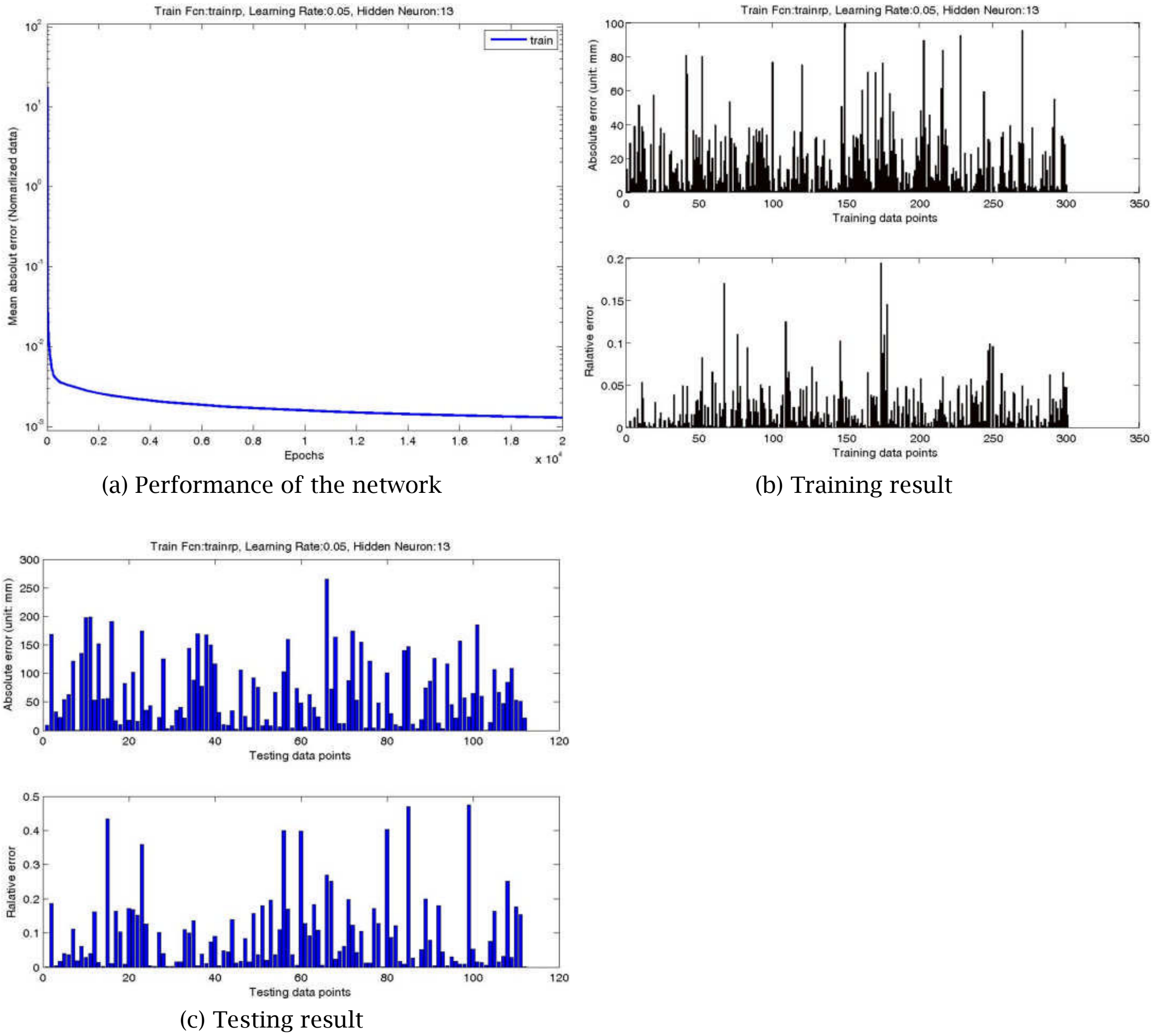
Tab.2:NN Parameters determined by experiment to have the best performance.

The performance curve (*MAE* vs. epochs) of the chosen neural network is shown in Fig. 7(a). The absolute error (*AE*) and relative error (*RE*) for all the training samples is shown in Fig. 7(b). After training, testing of the neural network is carried out using the testing examples collected from the 5 testing surfaces. Please note the testing inputs have been normalized using the same setting as the training data. The *AE* and *RE* for testing samples is given in Fig. 7(c). Some of the training and testing results are summarized in Tab. 3. Out of the 121 testing examples, only 10 have *RE* beyond 20%.

Parameters	*MAE*	*MRE*	*Max AE*	*Max RE*	*Samples with RE>20%*
Training	19.19 mm	0.017	99.2 mm	0.152	N/A
Testing	84.25 mm	0.143	275.8 mm	0.472	10/112

Tab. 3: Training and testing results.

It is observed that the tool-paths of the samples with *RE*>20% are relatively short. They all fall inside the range [54.2, 1024.5] while the longest tool-path in the training set stands at 30975.2 mm. This is probablybecause during the training, the weights and bias of the network are updated based on the *MAE* of the whole training set. Shorter tool-paths tend to have less influence as they are more likely to produce smaller *AE*, even though the corresponding*RE*can be quite big. However, this shortcoming will have limited effect on our search for the optimal cutter combination. As we have put a size limit on *eARs*, there cannot be many machining regions with very short tool-paths. Besides, these short tool-paths will only take up a small percentage of the overall tool-path length. Therefore we believe the resultant inaccuracies will be quite limited.



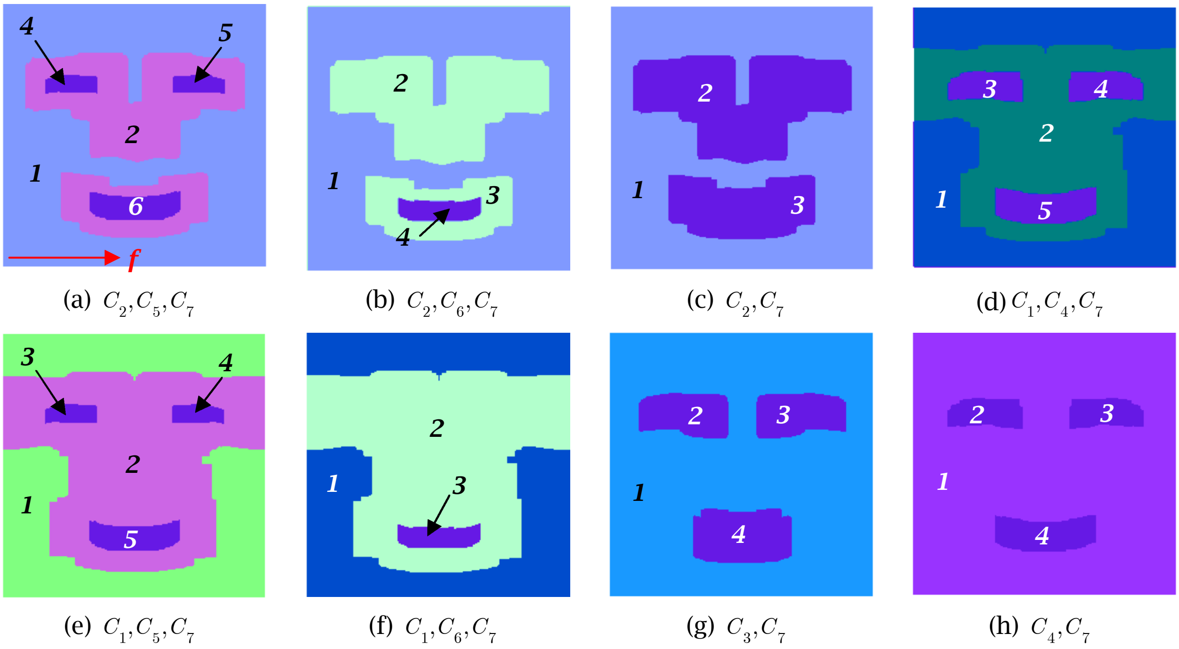
(a) Performance of the network

(b) Training result

(c) Testing result

Fig. 7: Training and testing of the neural network.

5 AN APPLICATION EXAMPLE



(a) C_2, C_5, C_7 (b) C_2, C_6, C_7 (c) C_2, C_7 (d) C_1, C_4, C_7

(e) C_1, C_5, C_7 (f) C_1, C_6, C_7 (g) C_3, C_7 (h) C_4, C_7

Fig. 8: Machining regions for all candidate cutter sets.

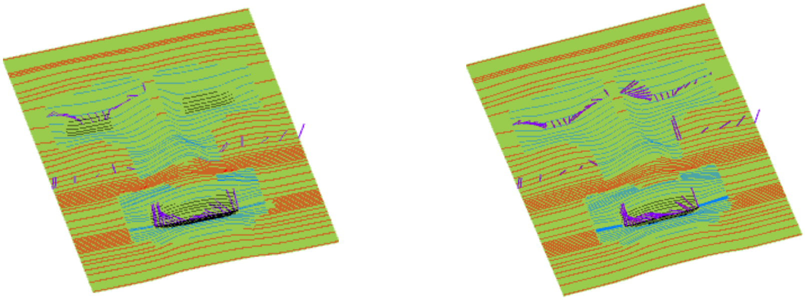
The 'FACE' shown in Fig. 2(a) is used as the testing surface. The cutter library given in Tab. 1 is used. After accessibility checking, the only accessible cutter in the cutter library is identified as C_7. The surface ratio limit for an eAR is set as 20%. Eight candidate cutter sets are identified based on this ratio, as given in Tab. 4. Their machining region allocations are given in Fig. 8.

CSet	Reg. #	Reg. 1	Reg. 2	Reg. 3	Reg. 4	Reg. 5	Reg. 6	Total	***Rank***
C_2, C_6, C_7	*Actual*	*3438.7*	*1481.8*	*592.8*	*280.1*	*N/A*	*N/A*	*5793.4*	***1***
	NN	*3647.1*	*1603.6*	*697.1*	*314.7*	*N/A*	*N/A*	*6262.5*	*2*
C_2, C_5, C_7	*Actual*	*3438.7*	*1045.8*	*554.9*	*214.3*	*226.8*	*321.6*	*5808.5*	*2*
	NN	*3647.1*	*942.1*	*694.4*	*184.5*	*164.8*	*328.2*	*5961.1*	***1***
C_2, C_7	*Actual*	*3438.7*	*2047.2*	*1035.5*	*N/A*	*N/A*	*N/A*	*6521.4*	*3*
	NN	*3647.1*	*2388.5*	*1217.4*	*N/A*	*N/A*	*N/A*	*7253.0*	*4*
C_1, C_4, C_7	*Actual*	*1379.3*	*4833.7*	*235.4*	*251.2*	*335.1*	*N/A*	*7034.6*	*4*
	NN	*1269.0*	*4547.5*	*195.5*	*275.5*	*286.4*	*N/A*	*6573.9*	*3*
C_1, C_5, C_7	*Actual*	*1379.3*	*5915.7*	*214.3*	*226.8*	*321.6*	*N/A*	*8057.7*	*5*
	NN	*1269.0*	*6510.2*	*244.5*	*284.8*	*198.2*	*N/A*	*8506.7*	*5*
C_1, C_6, C_7	*Actual*	*1379.3*	*6656.9*	*280.1*	*N/A*	*N/A*	*N/A*	*8316.3*	*6*
	NN	*1269.0*	*7330.8*	*310.5*	*N/A*	*N/A*	*N/A*	*8910.3*	*7*
C_3, C_7	*Actual*	*7225.4*	*438.3*	*415.1*	*754.7*	*N/A*	*N/A*	*8833.5*	*7*
	NN	*7308.2*	*395.2*	*401.2*	*690.1*	*N/A*	*N/A*	*8794.7*	*6*
C_4, C_7	*Actual*	*9053.1*	*235.3*	*251.2*	*335.1*	*N/A*	*N/A*	*9874.7*	*8*
	NN	*10007.1*	*195.5*	*275.5*	*286.4*	*N/A*	*N/A*	*10764.5*	*8*

Tab.4: Tool-paths lengths from actual data and prediction by NN (unit: mm).

The cutting direction *f* is selected to be along one of the boundaries of the 'FACE', as shown in Fig. 8(a). Algorithm 1 is used to generate the inputs to the neural network. After this, the neural network trained in section 4.2 is taken to predict the tool-path lengths for all the machining regions of each candidate cutter set. The overall tool-path length for a cutter set is calculated as the sum of tool-path lengths from all the machining regions. Finally, tool-path generation is carried out along the cutting direction f to obtain the accurate lengths of the tool-paths. The results from actual tool-path generation and the prediction made by NN are given in Tab. 4.

From Tab. 4, it can be seen theneural network can predict the tool-path length with a certain level of accuracy. This level of accuracy may not be enough to identify the cutter set with the shortest tool-path length every time, but is enough to make sure the one that gets selected is one of the best candidates. In our example, the actual tool-path length of optimal cutter set (C_2, C_6, C_7) is quite close to the one that gets selected by NN (C_2, C_5, C_7). Finally, to complete our study, the tool-paths of these two cutter sets are given in Fig. 9. Please note for a clearer view, the tool-postures for only a few tool-paths are plotted.



(a) Tool-paths for C_2, C_5, C_7 (by NN) (b) Tool-paths for C_2, C_6, C_7 (optimal set)

Fig. 9: Tool-paths for optimal cutter set and optimal set predicted by NN.

6 CONCLUSIONS

In this paper, a new method for multi-cutter selection in 5-axis finish machining of sculptured surfaces is proposed. The method is based on an accessibilitychecking method developed in our earlier work, which provides the cutter's accessibilityinformation in the form of A-maps. Based on the cutter's A-maps, the accessible regions for each cutter can be identified. Subsequently, the candidate cutter sets for finishing the entire surface areconstructed. To predict the total tool-path length for the candidate cutter sets, a neural network is developed with carefully chosen inputs. Finally the cutter set that produces the shortest overall tool-path length is selected as the optimal cutter set. The neural network is trained extensively and tested with samples that have never been exposed to it before. Testing results prove that our method can produce satisfactory accuracy for multi-cutter selection.

REFERENCES

[1] Arya, S., Cheng, S.; Mount, D.: Approximate algorithm for multiple-tool milling, International Journal of Computational Geometry and Applications, 11(3), 2001, 339–72. doi:10.1142/S0218195901000535

[2] Chen, Y. H.;Lee, Y. S.; Fang, S. C.: Optimal cutter selection and machining plane determination for process planning and NC machining of complex surfaces, Journal of Manufacturing Systems, 17(5), 1998, 371-388.doi:10.1016/S0278-6125(98)80004-6

[3] Choi, B.K.;Jerard, R.B.: Sculptured surface machining: theory and applications. 1998, Boston: Kluwer Academic Publishers

[4] D'Souza, R. M.;Sequin, C.; Wright, P.K.: Automated tool sequence selection for 3-axis machining of free-form pockets, Computer-AidedDesign, 36(7), 2004, 595-605. doi:10.1016/S0010-4485(03)00137-4

[5] Jensen, C. G.; Red, W. E.; Pi, J.: Tool selection for five-axis curvature matched machining, Comput-Aided Design, 34(3), 2002, 251-266.doi:10.1016/S0010-4485(01)00086-0

[6] Lee, Y. S.; Chang, T. C.: Automatic cutter selection for 5-axis sculptured surface machining, International Journal of Production Research, 34(4), 1996, 977-998. doi:10.1080/00207549608904946

[7] Li, H. Y.; Zhang, Y. F.: A geometric method for optimal multi-cutter selection in 5-axis finish cut of sculptured surfaces, IEEE International Conference on Automation and Logistics, 2008, 153-158.doi:10.1109/ICAL.2008.4636137

[8] Li, L. L.; Zhang, Y. F.: Cutter selection for 5-axis milling of sculptured surfaces based on accessibility analysis, International Journal of Production Research, 44(16), 2007, 3303-3323. doi:10.1080/00207540500444720

[9] Li, L. L.; Zhang, Y. F.: An Integrated Approach towards Process Planning for 5-axis Milling of Sculptured Surfaces based on Cutter Accessibility Map, Computer-aided Design and Applications, 3(1~4), 2006, 249-258.

[10] Park, S.C.;Choi, B. K.: Boundary extraction algorithm for cutting area detection, Computer-Aided. Design. 33(8), 2001, 571-579.doi:10.1016/S0010-4485(00)00101-9

[11] Sun, G.; Sequin, C. H.; Wright, P. K.:Operation decomposition for freeform surface features in process planning, Computer-Aided Design, 33(9), 2001, 621–636.
doi:10.1016/S0010-4485(01)00068-9

[12] Yao, Z.; Gupta, S. K.;Nau, D. S.: Algorithms for selecting cutters in multi-part milling problems, Computer-AidedDesign, 35(9), 2003, 825-839.doi:10.1016/S0010-4485(02)00110-0

[13] Yoon, J. H.;Pottmann, H.; Lee, Y. S.: Locally optimal cutter positions for 5-axis sculptured surface machining, Computer-Aided Design, 35(1), 2003, 69-81.
doi:10.1016/S0010-4485(01)00176-2