Progressive Segmentation for MRR-Based Feed-Rate Optimization in CNC Machining

Ka-Chun Chan[1] and Charlie C. L. Wang[1†]

***Abstract*— Keeping a constant cutting force in CNC machining is very important for obtaining better stability of cutting operation and improving topography, texture and geometry of the machined surface. This paper presents a feed-rate optimization approach based on *Material Removal Rate* (MRR). Given a tool-path with predefined feed-rates, the geometry of raw material, and the shape of cutter, the histogram of MRR in very fine resolution can be efficiently computed by using a GPU-based geometric modeling kernel. Starting from the evaluation given on the finest histogram of MRR, error-controlled subdivision algorithms are developed to progressively segment the tool-path into user-specified number of sub-regions. Different feed-rates are assigned to different sub-regions so that nearly constant MRR can be achieved while keeping the shape of the given tool-path unchanged. Experimental tests taken on real examples verify the effectiveness of this method.**

I. INTRODUCTION

Computer Numerical Control (CNC) machines that were first used in the 1950s now have been widely used in the industry to take mass production. With the help of CAD/CAM software, CNC machining process can be well planned beforehand and its quality and efficiency can be predicted precisely. A lot of researches have been taken to compute the tool-paths of CNC machining automatically so that they can be used to control the movement of cutters to form a designed shape by metal cutting. Recently, more and more researchers start to work on adjusting the feed-rate of cutter on a given tool-path to further optimize the machining result (ref. [1]). Different control parameters such as chip thickness, *Material Removal Rate* (MRR), maximal acceleration, and resultant forces have been used to optimize feed-rates. As lack of an efficient method to evaluate the Boolean operations in massive number of times, MRR-based method is only employed to adjust the feed-rate at a coarse level in the past (e.g., [2]). With the help of a geometric modeling kernel running on GPUs [3], we develop a new MRR-based technique that is able to optimize the feed-rates at any user specified level of details.

A. Problem Statement

During a machining process, MRR plays a very important role to the product's quality, the tool life and the process efficiency. MRR refers to the volume of workpiece being removed by the moving cutting tool in a unit time. Low MRR results in low efficiency of machining process. In the contrary, high MRR may result in bad surface finishing and internal stresses can be formed over the production. Tools can be broken in severe case. Therefore, maintaining an optimal MRR through optimizing the feed-rate of tool-path is inevitable in a good machining process. The term *feed-rate* here refers to the feeding speed of a cutting tool. The problem becomes more critical in some industrial applications, where the process of extrusion is applied on the material of workpiece with a large cutter (see Fig.1 for an example). A histogram of highly varied MRR can be observed, where higher MRR indicates a higher force applied on the cutter. Variation of such forces generates unwanted impact on the spindle of a machine and therefore results in deflection on the machined part. Based on this reason, we derive the first objective of MRR-based optimization of feed-rates.

Objective I: The MRR, $R(t)$, is controlled within $[R_{\min}, R_{\max}]$ during the machining.

The target range of MRR is specified by users according to the materials of the workpiece and the cutter as well as the stiffness of the CNC machine's spindle. In practice, they are determined via experimental tests. MRR with bounded variation during machining is achieved through adjusting the feed-rates of cutters. However, changing the feed-rates of a cutter too frequently within one tool-path such as what is shown in Fig.1 will easily cause the dynamic problems of a machine. Moreover, the software of some CNC machine limits the lines of G-code to be executed. Therefore, we place a second objective on our feed-rate optimization.

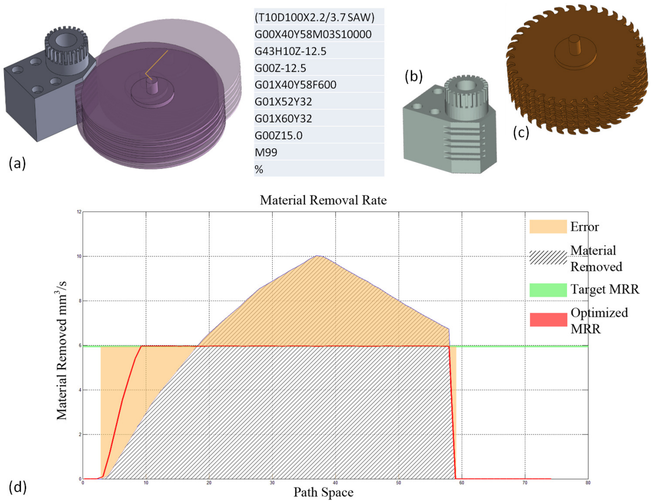
Objective II: While achieving a bounded MRR, $R(t) \in [R_{\min}, R_{\max}]$, the number of variations on feed-rates must be minimized.

To meet the criteria placed in these objectives, we develop a progressive segmentation approach in this paper. First, the histogram of MRR at very fine resolution is evaluated with the help of GPU-based solid modeling kernel [3]. The feed-rate scheduling to meet the requirement of $R(t)$ on a target MRR can then be taken at the finest resolution of tool engagement. To meet the demand on limited times of feed-rate adjustment, we develop a hybrid subdivision algorithm to segment the given tool-path into sub-regions and assign different feed-rates to each sub-regions. The subdivision is taken in a greedy manner with controlled variation of MRR.

*This work was partially supported by HKSAR Innovation and Technology Fund: ITS/247/11. Authors would like to thank Belton Technology Group for providing all the experimental tests and technical support.

[1] K.-C. Chan and C.C.L. Wang are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong

[†] Corresponding Author (Email: cwang@mae.cuhk.edu.hk)



```
(T10D100X2.2/3.7 SAW)
G00X40Y58M03S10000
G43H10Z-12.5
G00Z-12.5
G01X40Y58F600
G01X52Y32
G01X60Y32
G00Z15.0
M99
%
```

Fig. 1. An example machining process of extrusion by a large cutter: (a) the workpiece, the cutter, the tool-path (in blue) and G-code used in the simulation, (b) the resultant workpiece, (c) the shape of cutter, and (d) the MRR of this machining step changing significantly during the engagement of cutter – where the improved MRR with optimized feed-rates is displayed in red.

B. Related Works

To optimize the feed-rates of a tool-path in order to obtain enhanced performance and quality, the existing approaches can be classified into two groups: *online* adaptive-control based and *offline* re-scheduling based (ref. [1]). Representative works are respectively reviewed below.

Online adaptive control-based optimization refers to the methods using instant information (such as spindle speed, vibration, cutting force etc.) over the machining process to adjust the feed-rate simultaneously. Different types of data are acquired by attaching sensors on CNC machines, cutting tools or workpieces. Different methods have been proposed to optimize feed-rates with reference to data acquired in different ways. For example, a fuzzy adaptive control algorithm is applied in [4] with instant cutting force and vibration data. In [5], a dynamic characteristic-based fuzzy adaptive control algorithm is presented to avoid the influence of cutting force's variation, where the cutting force is indirectly evaluated in real-time by monitoring the motorized spindle current. This kind of online optimization approaches must have additional setup attached to each CNC machine and the cost of such setup is high. Moreover, as the approaches need to inference the control system of CNC machines, it can only be applied on CNC system with open architecture.

Offline re-scheduling is more widely employed to generate an optimized tool-path based on simulation. Comparing to the online approaches, offline methods are flexible and versatile. No additional device except a computer is required. Three different models were used in offline methods including volumetric models, vector-force models, and rule-based models. Rule-based models cannot precisely adjust feed-rates according to the shape variation during the machining process. Therefore, we only discuss volumetric models and force models here. Generally speaking, researchers in the past consider that the force-based model gives more precise prediction comparing to the volumetric model. However, force model is not general as various milling forces can be produced under different cutting conditions even with the same MRR (e.g., [6]–[9]). They are mainly developed for the purpose of surface finishing. The force prediction is difficult to be modeled when the machining process is extrusion with large volume of material removal as shown in Fig.1.

The feed-rate scheduling functions used in commercial CAD/CAM software are mainly based on MRR (ref. [1]). Average instead of instant MRR is usually employed to predict the cutting forces. As a result, feed-rates are only adjusted at very rough levels. Volumetric-model based offline optimization relies on the simulation of CNC machining, where different representatives have been employed for workpieces and cutters: image-space representation [2], [10], boundary representation [11] and space-decomposition based methods using voxel, octree and distance-field [12]–[14]. Benefited from a recently developed GPU-based geometric modeling kernel [3] based on the representation of Layered Depth-Normal Images (LDNI), MRR at very fine level can be obtained efficiently. For example, the MRR in Fig.1 can be obtained in 109 seconds on a PC equipped with graphics-accelerated hardware at consumer level. As a result, the feed-rate can be controlled at very fine resolution. Starting from the MRR at finest level, progressive segmentation algorithms are developed to generate optimized feed-rates meeting the demands of aforementioned objectives.

C. Contribution

The major technical contribution of this paper comes from a new segmentation algorithm for feed-rate re-scheduling in CNC-machining, which significantly improves the quality of production by offline optimization. Comparing to the online feed-rate optimization techniques that require an open architecture of numerical control, this technique can be more generally taken on different types of CNC systems – it therefore has higher industrial impact in practice. To the best of our knowledge, this is the first MRR-based optimization approach that can result in optimal feed-rates with user-specified number of adjustments.

II. MRR Evaluation

MRR at the finest resolution for a given tool-path is evaluated with the help of GPU-based solid modeling kernel [3]. Our study focuses on the 3-axis CNC machining although it can be generalized to handle 5-axis machining as well. As illustrated in Fig.1, the tool-path is actually a transformation function, $\mathbf{T}(t) \in \Re^3$, that can be illustrated as a 3D curve. We subdivide the whole trajectory into small intervals with length d (e.g., $d = 0.5mm$). These intervals are used to simulate the engagements of a cutting tool during the machining process. Without loss of generality, $\mathbf{p}_i$ is employed to denote the starting point of the i-th interval. That is, in this interval – also called the i-th engagement I_i, the center of cutter's frame moves from $\mathbf{p}_i$ to $\mathbf{p}_{i+1}$. There are n such intervals, which approximate the engagement of a cutter by cascaded transformations: $\mathbf{T}_{1,\dots,n}$ with $\mathbf{T}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$.

Given a 3D solid model $\mathcal{W}$ for the workpiece before machining and a 3D model $\mathcal{T}$ for the cutter, solid modeling operators are employed to evaluate the shape variation of the workpiece during machining. Note that, the 3D model of a cutter used in the simulation can be obtained by revolution around its self-rotating axis (e.g., the model shown in Fig.1(b)). The shape of $\mathcal{W}$ after k engagements of the cutter can be obtained by Boolean operators as

$$\mathcal{W}_k = \mathcal{W}_{k-1} \setminus \mathcal{S}_k(\mathcal{T}) \tag{1}$$

with $\mathcal{S}_k(\mathcal{T})$ denoting the swept volume of $\mathcal{T}$ from $\mathbf{p}_k$ to $\mathbf{p}_{k+1}$ and '$\setminus$' for subtraction operation of solids. When knowing the feed-rate of cutter in this engagement, the material removal rate R_k can be evaluated by

$$R_k = \frac{V(\mathcal{W}_{k-1}) - V(\mathcal{W}_k)}{t_k}, \tag{2}$$

where t_k is the time spent on moving $\mathcal{T}$ from $\mathbf{p}_k$ to $\mathbf{p}_{k+1}$, and $V(\cdots)$ represents the volume of a 3D model.

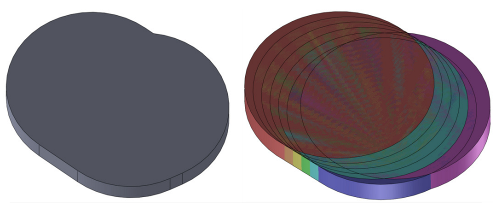
Computing a general swept volume is hard. Therefore, approximate methods are introduced to evaluate the swept volume more effectively (ref. [13], [15]). In our application for feed-rate optimization w.r.t. MRR at a fine resolution, the number of intervals is large. Boolean operations on 3D models need to be intensively applied. Evaluating exact Boolean operation on complex 3D models is time-consuming. Therefore, we conduct an alternative strategy to compute approximate MRR. First of all, Boolean operations are taken on a sampling-based representation – Layered Depth-Normal Images (LDNI), which is an extension of ray-rep. With the help of LDNI representation, the computation in 3D Boolean operations is degenerated into logic operations in 1D and can be performed in a highly parallel manner. This well fits the architecture of modern graphics hardware. As a result, Boolean operations on LDNI solids can be computed very efficiently (ref. [3]). Furthermore, when n is large (i.e., a small distance d is employed in our experimental tests), we take the following approximation of $\mathcal{S}_k(\mathcal{T})$ that

$$\mathcal{S}_k(\mathcal{T}) = (\mathbf{p}_{k+1} - \mathbf{p}_1)\mathcal{T} \cup (\mathbf{p}_k - \mathbf{p}_1)\mathcal{T}. \tag{3}$$

This results in a simplified method to evaluate the shape of $\mathcal{W}$ after k engagements as

$$\widetilde{\mathcal{W}}_k = \widetilde{\mathcal{W}}_{k-1} \setminus ((\mathbf{p}_{k+1} - \mathbf{p}_1)\mathcal{T}). \tag{4}$$

An experimental tests is taken to verify the accuracy of this simplification (see the illustration in Fig.2). We use this method to compute the shape of $\bigcup_{k=1}^{n}((\mathbf{p}_{k+1} - \mathbf{p}_1)\mathcal{T})$, and it is compared with the analytically computed swept volume of the whole trajectory (i.e., $\bigcup_{k=1}^{n} \mathcal{S}_k(\mathcal{T})$). LDNI representation with high resolution is employed in the tests. Very small error is observed from the statistics shown in Fig.2. In summary, this simplification is accurate enough to be used in the evaluation of MRR at fine resolutions.



d	LDNI-based Boolean		Analytic	Err. (%):
	LDNI Res.	Volume (V_a)	Volume (V_b)	$\frac{\lvert V_a - V_b \rvert}{V_b}$
2.0	1024^2	30785.7	30801.4	0.0509%
1.0	1024^2	30786.5	–	0.0482%
0.8	1024^2	30786.8	–	0.0473%
0.5	1024^2	30787.0	–	0.0467%

Fig. 2. An experimental test to verify the accuracy of the simplified evaluation of MRR by LDNI representation: (left) the analytically computed swept volume [16] and (right) the illustration for cascaded Boolean operations by LDNI-based computation. The volume in LDNI-representation is obtained by the average of LDNI solids in x-, y- and z-directions.

III. FEED-RATE OPTIMIZATION

Using the aforementioned method, MRR at fine resolution can be obtained efficiently. This section presents the algorithms for re-scheduling the feed-rates so that a target MRR can be achieved during the engagements of tools. Meanwhile, the number of feed-rate adjustments is minimized.

A. Re-Scheduling at Fine Level

The purpose of feed-rate optimization is to obtain a smooth and nearly constant MRR throughout the whole machining process by altering the feed-rates accordingly. At the instant that the cutting tool is removing small or no volume of material, we speed up the cutting tool's engagement to improve the efficiency. When the cutter is removing a large volume of material, the movement of cutter is slowed down to reduce impact on both the workpiece and the cutter. This can be realized easily when it is allowed to adjust the feed-rate F during each interval of our MRR evaluation. In general, the average MRR in the i-th interval is V_i/t with $V_i = V(\mathcal{W}_{i+1}) - V(\mathcal{W}_i)$ being the volume of removed material. Here t can be obtained from the length of an interval divided by the speed of tool's movement as $t = d/(\frac{F}{60})$, and the feed-rate is scaled by 60 as its unit in G-code is mm/min. Therefore, to meet a target MRR $\bar{R}$, the feed-rate F_i in each interval is adjusted to

$$F_i = \frac{60d}{V_i} \cdot \bar{R}. \tag{5}$$

However, such kind of feed-rate scheduling taken at the fine level results in too many times of feed-rate adjustments to meet the demand described in *Objective II*.

B. Re-Scheduling by Groups

The divide-and-conquer strategy is applied to re-schedule the feed-rates to satisfy the demands in both objectives. Limited number of feed-rates are given to minimize the resultant MRR in each interval with reference to the target MRR. This can be considered as a segmentation problem to subdivide all the n intervals into m groups with $m \ll n$. The intervals in each group Π_j are assigned with a constant feed-rate $F(\Pi_j)$. For $i \in \Pi_j$, when $F(\Pi_j)$ is used as the feed-rate of cutter, MRR in the interval is

$$R_i = \frac{F(\Pi_j) V_i}{60d}. \tag{6}$$

$E_i = \left(R_i - \bar{R}\right)^2$ defines the MRR error on the i-th engagement. Given a segmentation with groups $\{\Pi_j\}$, the total MRR error on the whole tool-path is then defined as

$$E(\{\Pi_j\}) = \sum_i E_i = \sum_i \left(\frac{F(\Pi_j) V_i}{60d} - \bar{R} \right)^2. \tag{7}$$

This actually defines the L^2-norm between the current MRR and the target. Minimizing $E(\{\Pi_j\})$ results in the minimal deviation of MRR in each engagement w.r.t. $\bar{R}$. From

$$\partial E(\{\Pi_j\}) / \partial F(\Pi_j) = 0,$$

we can derive the optimal value of feed-rates in each group as

$$F(\Pi_j) = 60 d \bar{R} \cdot \frac{\sum_{i \in \Pi_j} V_i}{\sum_{i \in \Pi_j} V_i^2}. \tag{8}$$

Note that, in some CNC machines, the values of feed-rates are quantized into limited levels. In such cases, $F(\Pi_j)$ obtained from Eq.(8) is projected onto the nearest value in quantization.

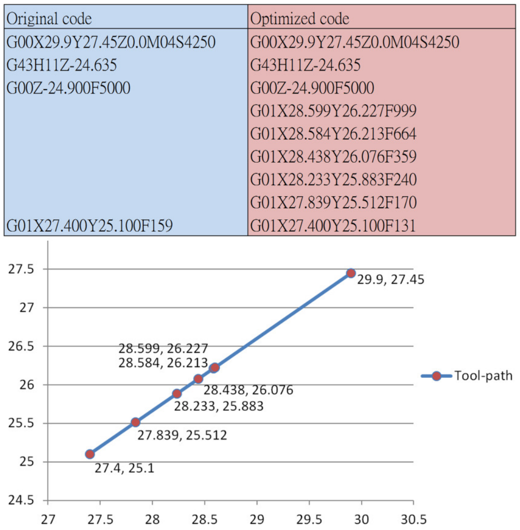
Now the left problem is how to find a minimal segmentation with $R_{\min} \leq R_i \leq R_{\max}$. At the same time, the feed-rates must also be controlled within a user specified range $[F_{\min}, F_{\max}]$, which are constrained by the performance of CNC machines. We first group the intervals according to the engagements specified in G-code of the input tool-path. Each group is assigned with a temporary feed-rate by Eq.(8). In the case that the determined $F(\Pi_j)$ is out of the range $[F_{\min}, F_{\max}]$, a nearest feed-rate in the allowed range is assigned to $F(\Pi_j)$. Next we further optimize the feed-rates by subdividing the groups into smaller sub-groups iteratively. The objective of this iteration is to make $R_{\min} \leq R_{i=1,\cdots,n} \leq R_{\max}$ be satisfied using limited number of groups and feed-rates.

By means of subdividing a group between the engagements I_i and I_{i+1}, we insert the point $\mathbf{p}_{i+1}$ as an intermediate location in the G-code of tool-path. As demonstrated in Fig.3, one engagement in the input G-code has been refined into five engagements with different feed-rates.

C. Subdivision by Exhaustive Check

A greedy subdivision algorithm is developed to subdivide one existing group of engagement into two sub-groups. Specifically, the subdivision process mainly involves three steps:

1) Selecting the next group of engagement to divide;
2) Locating a best place to conduct the subdivision;
3) Assigning feed-rates to the newly created groups of engagement.



Original code	Optimized code
G00X29.9Y27.45Z0.0M04S4250	G00X29.9Y27.45Z0.0M04S4250
G43H11Z-24.635	G43H11Z-24.635
G00Z-24.900F5000	G00Z-24.900F5000
	G01X28.599Y26.227F999
	G01X28.584Y26.213F664
	G01X28.438Y26.076F359
	G01X28.233Y25.883F240
	G01X27.839Y25.512F170
G01X27.400Y25.100F159	G01X27.400Y25.100F131

Fig. 3. Feed-rate of the input tool-path has been rescheduled by refining one engagement into five – see the refined tool-path and the corresponding G-code.

We repeatedly run these three steps until either of the following two terminal conditions have been reached:

- MRRs in all engagements have fallen in the range of $[R_{\min}, R_{\max}]$;
- The number of groups has reached the maximal allowed number η.

Here the second terminal condition would be favorable to the CNC machines that cannot process many blocks of G-code at a time. Moreover, having too many times of change in feed-rate will result in bad dynamic performance during machining.

As a greedy algorithm, a group possessing the highest MRR error on any of its interval is selected in the first step of each iteration. Specifically, the group to be refined is

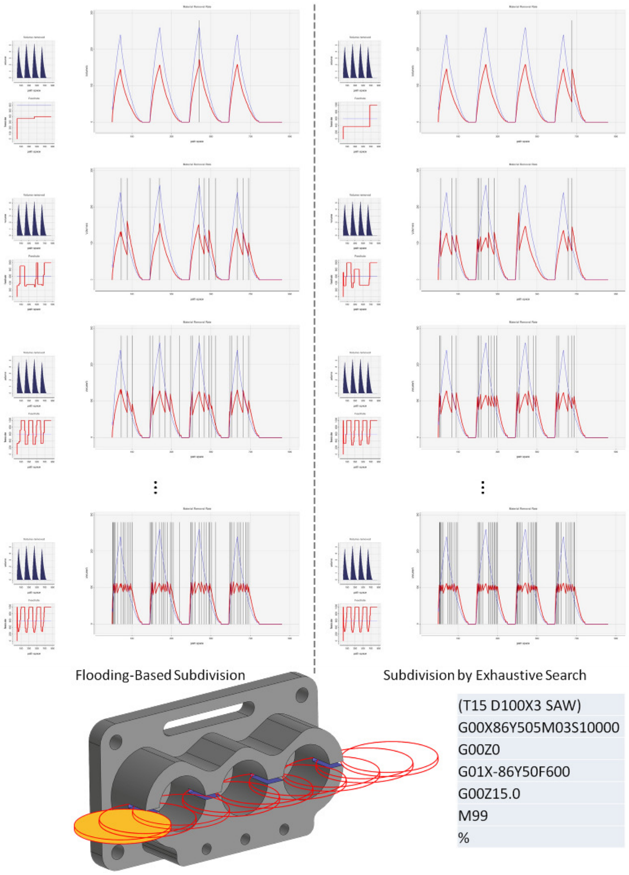
$$\Pi_{act} = \arg\max_{\Pi_j} \{\forall i \in \Pi_j \mid E_i\}. \tag{9}$$

This selection in fact is taken according to the L^∞-norm on the MRR error.

In the second step of subdivision, among all engagements $I_{s,\cdots,e}$ inside Π_{act}, we need to generate a partition to separate them into two sub-groups: I_s to I_l (denoted by $\Pi_{act}^L[l]$) and I_{l+1} to I_e (denoted by $\Pi_{act}^R[l]$). For a given l, the optimal feed-rates can be obtained by Eq.(8) for $\Pi_{act}^L[l]$ and $\Pi_{act}^R[l]$ respectively. With which, the resultant MRR error according to the subdivision at l is

$$\begin{aligned} E_{sub}(\Pi_{act}, l) &= \sum_{i=s}^{l} \left(\frac{F(\Pi_{act}^L[l]) V_i}{60d} - \bar{R} \right)^2 \\ &\quad + \sum_{i=l+1}^{e} \left(\frac{F(\Pi_{act}^R[l]) V_i}{60d} - \bar{R} \right)^2 \end{aligned} \tag{10}$$

Among all the possible choices of l, the one leading to a minimal $E_{sub}(\Pi_{act}, l)$ is employed to conduct the subdivision. The right column of Fig.4 presents the progressive results generated by this algorithm of exhaustive search.



```
(T15 D100X3 SAW)
G00X86Y505M03S10000
G00Z0
G01X-86Y50F600
G00Z15.0
M99
%
```

Fig. 4. Comparison of subdivisions based on different strategies. From top to bottom, the progressive results after taking 1, 10, 20 and 50 refinements, where the red curves show the improved MRR. The corresponding feed-rates are also given in the left-bottom corner of each report (in red). The workpiece and the input G-code are shown at the bottom.

D. Subdivision by Flooding

In the algorithm introduced above, exhaustive search is taken to locate a best place to conduct the subdivision. The algorithm has the complexity of $O(n^2)$ for a group with n engagements. A more efficient flooding algorithm is developed here to get a good location of subdivision with linear complexity – $O(n)$. Different from the subdivision using the exhaustive search strategy, this flooding algorithm is based on the heuristic that bounding the range of MRR variation in both left and right sub-groups leads to a good partition. With this partition and the optimal feed-rates obtained from Eq.(8), the MRR error can also be reduced promptly.

Starting from the most-left engagement I_s and the most-right engagement I_e, two sub-groups are progressively enlarged where the one introducing smaller MRR variation will have higher priority to expand. Detail steps for generating the partition are listed below.

1) Initialization by $\Pi_{act}^{L} = \{I_s\}$ and $\Pi_{act}^{R} = \{I_e\}$;
2) Assign $L_{\min} = L_{\max} = R_s$, $R_{\min} = R_{\max} = R_e$;
3) Evaluate the cost of expanding left sub-group as
$$C_L = \max\{R_{\max} - R_{\min}, \\ \max(L_{\max}, R_{s+1}) - \min(L_{\min}, R_{s+1})\};$$

Fig. 5. More than thousands of CNC machines are all suffered from the problem of unwanted impacts generated by the significant variation of MRR.

4) Evaluate the cost of expanding left sub-group as
$$C_R = \max\{L_{\max} - L_{\min}, \\ \max(R_{\max}, R_{e-1}) - \min(R_{\min}, R_{e-1})\};$$
5) When $C_L < C_R$, adding I_{s+1} into Π_{act}^{L}, updating $L_{\min}$ & $L_{\max}$ and letting $s = s + 1$;
6) If $C_L \geq C_R$, adding I_{e-1} into Π_{act}^{R}, updating $R_{\min}$ & $R_{\max}$ and assigning $e = e - 1$.
7) Go back to step 3) until $s = e - 1$.

The left column of Fig.4 illustrates the progressive results generated by this flooding algorithm.

E. Hybrid

We compare the converging speed on the MRR errors of the above two subdivision strategies. As shown in Table I and Fig.4, the flooding based method converges in the same trend as exhaustive search but slightly slower. Based on this observation, a hybrid approach is employed in our implementation. When the number of engagements in a group is more than z (e.g., $z = 30$ is used in our tests), the flooding based subdivision is employed. Once groups becomes small, the exhaustive search is conducted to find the best refinement.

TABLE I
COMPARISON OF DIFFERENT SUBDIVISION STRATEGY

No. of Groups	MRR Error: $E(\{\Pi_j\})$	
	Flooding	Exhaustive Search
11	1,798,634.0	1,604,388.1
21	1,283,127.0	1,143,502.4
31	1,147,297.1	1,111,366.8
41	1,101,741.4	1,081,905.8
51	1,084,741.2	1,073,660.6

IV. RESULTS AND CASE STUDY

The whole pipeline of our MRR-based feed-rate optimization approach is formed by 1) the GPU-based solid modeling kernel for evaluating the MRR at fine resolution and 2) the progressive segmentation approach for generating optimal feed-rates with limited number of variations. The GPU-based solid modeling kernel is implemented by C++ and nVIDIA CUDA Library, and source code of the kernel is publicly accessible (ref. [17]). The optimization algorithms are implemented by Processing [18]. Example results can be found in Figs.1 and 4.

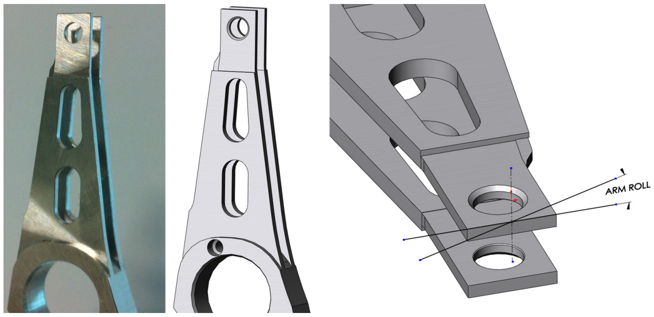


Fig. 6. Actuator-arm used in the case study – from left to right: the real part, the CAD model and the definition of arm-roll for verifying the deflection.

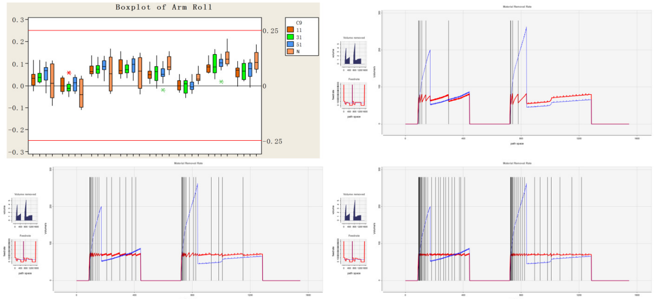


Fig. 7. Statistical arm-rolls with target MRR: $\bar{R} = 70$ generated by trials using optimized tool-path with different number of feed-rate variations. Here, 'N' denotes the input tool-path with a constant feed-rate. MRRs according to optimized feed-rates are also given.

Our case study is collaborated with a machining factory producing actuator-arms in hard-disk drives, which read/write data to disks. The quality of this actuator-arm is of paramount importance as small deflection on the actuator-arm will lead to unwanted scratches on the disk surface made by the magnetic head attached on the arm. Extrusion machining is intensively employed on more than thousand CNC machines in the factory (see Fig.5). All are suffered from the problems of unwanted impacts generated by non-uniform material removal rate. In our study, feed-rates in the last machining process of extrusion are considered. The deflection of actuator-arms is analyzed by measuring the arm-roll on the tip of an arm (see Fig.6 for an illustration).

In our tests, the target MRR $\bar{R}$ is assigned as 70. Experiments are taken on the optimized feed-rates with 11, 31 and 51 adjustments respectively. The statistical measurements of arm-rolls are shown in Fig.7 and compared with the machining result using constant feed-rate (denoted by 'N'). It can be observed that our optimization results in smaller deviation on arm-role among the product trials. Moreover, most trials obtained from the optimized feed-rates show tendency towards *zero* arm-roll.

V. CONCLUSION

In this paper, we present an approach to automatically optimize the feed-rates of an input tool-path according to the *material removal rate* (MRR). The objective of this optimization is to adjust MRR on a resultant tool-path as constant as possible. Another constraint on the number of feed-rate adjustments has also been considered in this approach by a progressive segmentation strategy. As a result, our method can effectively reduce the variation on MRR of a given tool-path with limited number of feed-rate adjustments. This has also been verified by a case study on real products. More experiments will be carried out to further verify the effectiveness of this approach on different machines and different materials. We plan to attach sensors on cutter to monitor the influence over different parameters after the feed-rate re-scheduling.

Only tool-paths of 3-axis CNC machining are considered in this paper. However, as long as the MRR at fine resolution can obtained, our feed-rate optimization algorithm can be generally applied to tool-paths of 4-axis and 5-axis CNC machining. This will be conducted in our future work.

REFERENCES

[1] M. Kurt and E. Bagci, "Feedrate optimisation/scheduling on sculptured surface machining: a comprehensive review, applications and future directions," *International Journal of Advanced Manufacturing Technology*, vol. 55, pp. 1037–1067, 2011.

[2] VeriCUT, http://www.cgtech.com, 2013.

[3] C. C. L. Wang, Y.-S. Leung, and Y. Chen, "Solid modeling of polyhedral objects by layered depth-normal images on the GPU," *Computer-Aided Design*, vol. 42, no. 6, pp. 535–544, 2010.

[4] F. Ridwan, X. Xu, and F. C. L. Ho, "Adaptive execution of an NC program with feed rate optimization," *International Journal of Advanced Manufacturing Technology*, vol. 63, no. 9-12, pp. 1117–1130, 2012.

[5] H. Liu, T. Wang, and D. Wang, "Constant cutting force control for CNC machining using dynamic characteristic-based fuzzy controller," *Shock and Vibration*, vol. 2015, 2015.

[6] G. Kim, P. Cho, and C. Chu, "Cutting force prediction of sculptured surface ball-end milling using z-map," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 2, pp. 277–291, 2000.

[7] L. Ning and S. C. Veldhuis, "Mechanistic modeling of ball end milling including tool wear," *Journal of Manufacturing Processes*, vol. 8, no. 1, pp. 21–28, 2006.

[8] R. S. Naserian, M. Sadeghi, and H. Haghighat, "Static rigid force model for 3-axis ball-end milling of sculptured surfaces," *International Journal of Machine Tools and Manufacture*, vol. 47, no. 5, pp. 785–792, 2007.

[9] C.-L. Tsai and Y.-S. Liao, "Prediction of cutting forces in ball-end milling by means of geometric analysis," *Journal of Materials Processing Technology*, vol. 205, pp. 24–33, 2008.

[10] B. K. Fussell, R. B. Jerard, and J. G. Hemmett, "Robust feedrate selection for 3-axis nc machining using discrete models," *Journal of manufacturing science and engineering*, vol. 123, no. 2, pp. 214–224, 2001.

[11] A. D. Spence, F. Abrari, and M. A. Elbestawi, "Integrated solid modeler based solutions for machining," in *Proceedings of ACM Symposium on Solid Modeling and Applications*, 1999, pp. 296–305.

[12] D. Jang, K. Kim, and J. Jung, "Voxel-based virtual multi-axis machining," *International Journal of Advanced Manufacturing Technology*, vol. 16, no. 10, pp. 709–713, 2000.

[13] K. Karunakaran, R. Shringi, D. Ramamurthi, and C. Hariharan, "Octree-based nc simulation system for optimization of feed rate in milling using instantaneous force model," *International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5-8, pp. 465–490, 2010.

[14] A. Sullivan, H. Erdim, R. N. Perry, and S. F. Frisken, "High accuracy NC milling simulation using composite adaptively sampled distance fields," *Computer-Aided Design*, vol. 44, no. 6, pp. 522–536, 2012.

[15] Y. J. Kim, G. Varadhan, M. C. Lin, and D. Manocha, "Fast swept volume approximation of complex polyhedral models," in *Proceedings of ACM Symposium on Solid Modeling and Applications*, 2003, pp. 11–22.

[16] SolidWorks, http://www.solidworks.com/, 2014.

[17] LDNICuda, http://ldnibasedsolidmodeling.sourceforge.net/, 2014.

[18] Processing, http://processing.org/, 2014.