International Journal of Innovative Computing, Information and Control
Volume **8**, Number **5(A)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3265–3278

IMPROVED TOOL POSITIONING IN 5-AXIS FLANK MILLING BY SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION

Chih-Hsing Chu, Ping-Han Wu and Hsin-Ta Hsieh

Department of Industrial Engineering and Engineering Management
National Tsing Hua University
No. 101, Section 2, Kuang-Fu Road, Hsinchu 30013, Taiwan
chchu@ie.nthu.edu.tw

Received January 2011; revised May 2011

Abstract. *This paper presents an improved tool positioning strategy in the 5-axis flank milling of a ruled surface by using simultaneous perturbation stochastic approximation (SPSA) techniques. The SPSA allows the near-global optimization of the machining error, even though the corresponding objective is not formulated as a closed-form representation. It adjusts all cutter locations of a tool path simultaneously, and reduces the induced deviation by not considering the CNC interpolation in the tool path planning. The resultant tool path produces a smaller error compared with the greedy approach that was adopted by several previous studies. This study provides a simple but systematic mechanism for machining error control in 5-axis flank milling.*

Keywords: Flank milling, 5-axis machining, Tool path planning, Optimization

1. **Introduction.** 5-axis flank milling has recently received notable attention because of its productivity and flexibility. This machining operation is commonly used to manufacture ruled geometry in the aerospace, automobile, mold, and energy industries. The cutting action occurs along the cutter peripheral, resulting in a higher material removal rate. The tool motion is highly complex with the two additional degrees of freedom, and the shaping capability is superior to that of traditional 3-axis machining. However, tool interference is difficult to eliminate, and control of the machining error is a challenging task.

As illustrated in Figure 1, a cylindrical milling cutter moving along the boundary curves of a ruled surface induces tool interference when the contact region is not locally developable [1]. Flank milling produces deviation on the machining surface with respect to design specification, except for simple geometries, such as cylindrical and conical surfaces. The deviation is referred to as the machining error. Several studies developed tool positioning strategies to reduce the error in five-axis flank milling. These strategies are classified into two categories. The first class [2-10] approximates the tool path with a set of discrete cutter locations. The optimized methods are independently applied to individual cutter locations for reducing the error. The second class [11-13] models the volume that is swept by continuous tool movement as an envelope surface. The deviation is estimated by comparing the envelop surface with the design surface. The result helps distinguish the quality of the tool paths that were generated from various methods.

Stute et al. [2] proposed a heuristics-based positioning method, in which the cutter was tangential to the boundary curves of the ruled surface and contacted the surface at the middle of the ruling line. However, the excessive undercut or overcut still occurred, particularly when the machining surface was twisted. Liu [3] proposed a heuristic algorithm that offsets two points at a quarter and three-quarters of the length of the ruling line

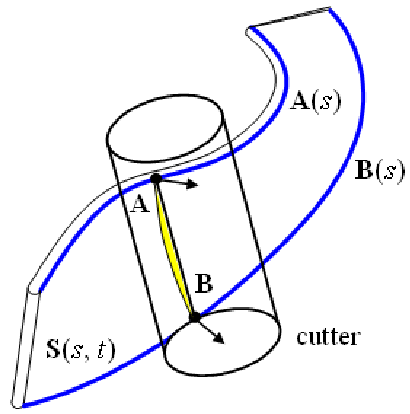


Figure 1. Tool interference in 5-axis flank milling caused by non-developability [1]

along the surface normal with the distance of a cutter radius. The tool orientation was defined by joining the offset points. Notable tool interference occurred near the middle of the machined surface. Bohez et al. [4] positioned the cutter to contact the ruling line at a point where the surface normal maintained an equal angle to the surface normals at the end points of the ruling. The maximal overcut occurred in the mid-curve of the ruled surface. Redonnet et al. [5] presented a strategy that positioned the cutter tangent to the ruled surface at three points. The optimal tool orientation was determined by simultaneously solving a system of seven transcendental equations. The robustness and computation efficiency prevented the strategy from practical use. Lee and Suh [6] proposed an interference-free positioning algorithm that adjusted the tool orientation by either rotating around, or offsetting along the surface normal at the middle of the ruling line. Their goal was to minimize the overcut volume at individual tool positions. Tsay and Her [7] developed a tool positioning method similar to that of [6]; however, they used different reference directions to offset and rotate the tool axis. Bedi et al. [8] placed the cutter tangent to the boundary curves of the ruled surface with the contact points on the boundary curves at the same parameter value. This method produced the maximal deviation in the middle area of the ruled surface. Based on this work, they developed a three-step optimization scheme for minimization of the deviation [9]. Senatore et al. [10] proposed a positioning method based on the analysis of various tool orientations in regard to the machining error. The cutter was positioned tangent to the boundary curves of the surface and touched the ruling line only at one point.

Other studies attempted to solve the tool positioning problem with the envelope surface approach. Lartigue et al. [11] defined the tool trajectory by constructing a ruled surface with two curves that moved along a specific rail. They displaced the control points of the curves to deform the envelop surface to minimize the deviation. Chiou [12] proposed two strategies (one strategy for rough milling and the other strategy for finish milling) to correct the tool position when the envelop surface collided with the design surface. Gong et al. [13] developed a three points offset (TPO) method based on the assumption that the deviation between the envelope surface that is generated by a cylindrical cutter and the design surface is equivalent to that between the offset surface of the latter and the tool axis trajectory surface.

Several problems are presented in these methods and thus limit their effectiveness. In the first class of studies, all of the "optimization" attempts form part of the approach that locally adjusted the tool position and orientation at distinct locations. The resultant tool path was not guaranteed to be optimal in terms of minimizing the total error on the machined surface. Our previous work [14] experimentally demonstrated that the deviation induced by not considering the CNC interpolation between cutter locations may be higher than the error reduced by the local adjustment of distinct cutter locations. The second class of studies computed the tool path based on heuristic algorithms. These studies did not offer analytical evidence for the effectiveness or robustness of the algorithms. The CNC interpolation complicated the studies by augmenting the approximation error of the envelope surface that was used in those methods. The surface construction process did not consider the actual interpolation motion. The main difficulty in optimizing the tool path is the formulation of the objective function, regardless of whether the discrete or continuous approaches are used. No closed-form representation exists for the machining error estimation. Consequently, the gradient-based search methods are not applicable. In practice, a tool path in five-axis milling comprises a large number of cutter locations. The optimization schemes involve significant computation in the search process.

Several previous studies adopted the heuristic approaches to reduce the machining error in 5-axis flank milling. Those approaches were applied to adjust the individual cutter locations independently, without considering the tool motion between the cutter locations or the resulting machining errors. This problem may be overcome by the simultaneous adjustment of all cutter locations. Thus, we propose the application of simultaneous perturbation stochastic approximation (SPSA) to the tool path planning in 5-axis flank milling. The SPSA simultaneously adjusts all cutter locations of a tool path, thereby overcoming the problem inherited from the greedy approach. Such a near-global optimization scheme reduces the total error on the machined surface from a holistic perspective. It lowers the inaccuracy that is induced by not considering the actual interpolated tool motion. The proposed method is particularly useful when the objective function is not analytically differentiable. The simulation results indicate that the proposed method is superior to the previous optimization-based methods and heuristic approaches in terms of error reduction. The SPSA-based tool positioning offers a simple, systematic, and effective method for machining error control in 5-axis flank milling of a ruled surface.

2. **SPSA-based Tool Positioning.**

2.1. **SPSA preliminaries.**

This study proposes an SPSA-based tool positioning strategy for 5-axis flank milling that simultaneously optimizes the distinct cutter locations of a tool path. The method consists of the following steps: first, a set of ruling lines with equal parametric intervals were generated from the surface to be machined. Given a

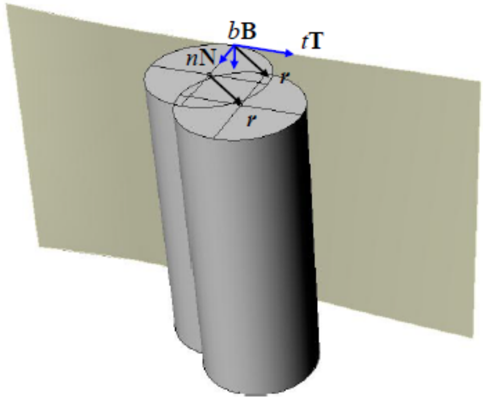


FIGURE 2. Illustration of the proposed tool positioning method

ruling, the initial orientation of the cutter was determined by linking the two points offset from the end points of the ruling along the surface normals with a distance of the cutter radius. The second step was to establish the Frenet frames at the end points of each ruling line. The Frenet frame consisted of the following three mutually orthogonal vectors: the normal N, the tangent T, and the bi-normal B. The frame served as a local coordinate system for adjusting the tool orientation at a cutter location. Three scalars, t, n, and b were introduced to control the adjustment magnitudes of the orientation along T, N, and B. These three scalars became the variables to be optimized in the SPSA method.

A direction vector for a point on the boundary curve was generated with the three unit vectors and scalars, as illustrated in Figure 2. It was expressed as (tT, nN, bB). The two center points of the cutter were translated with the corresponding direction vectors. Our aim was to determine an optimal set of scalars by using the SPSA method, which minimizes the machining error. This task is described as:

$$\text{Min } E(\theta) \tag{1}$$

where E is the total error produced on the machined surface; θ represents a set of 3-tuple (t, n, b).

In this method, we controlled the trajectory of a cylindrical cutter along a ruled surface by specifying two sets of 3-tuples. The previous study demonstrated that the adjustment of the tool axis along the surface normal produced a markedly greater effect on the machining error as compared to the adjustment along the tangent or the bi-normal [7]. Our preliminary test also indicated that the optimization result was sensitive to the variable n. Therefore, it was inadequate to optimize the three scalars at the same time because their sensitivities were not of the same order of magnitudes [15]. We chose not to include n in the optimization by considering it as a constant that was equal to the cutter radius. In this case, the total number of variables to be determined was 2N, where N was the number of the surface rulings (or the number of cutter locations) to be considered. This number influenced the convergence rate of the search process by SPSA. Compared to classic routing or scheduling problems, the tool path optimization involves a solution space of markedly higher dimensions.

2.2. **Machining error estimation with z-buffer method.** The machining error was estimated with the z-buffer method that was developed in the previous study [16]. It defined an approximate measure M of the function E in Equation (1). The exact estimation of E was highly complex. In this study, the estimation of the machining error consisted of four steps, as depicted in Figure 3. The design surface was first sampled in a discrete manner. At each sampling point, two straight lines extended along the positive and negative normal directions with a distance of the cutter radius. The lengths of these lines were updated after the cutter swept across them along a given tool path. The lines illustrated above and below the surface indicate the amount of undercut and overcut, respectively. We approximated the tool swept surface between two cutter locations at a finite number of interpolated tool positions. This approximation was similar to the actual tool motion generated by the CNC interpolation. The next step was to intersect the lines with the peripheral surface of the cutter. The machining error was calculated as the sum of the lengths of the trimmed straight lines.

2.3. **Simultaneous perturbation stochastic approximation.** The SPSA was first proposed by Spall [15] to solve the multivariate optimization problems, in which the objective functions are not differentiable but may be estimated by noisy measurements. It is efficient and relatively easy to implement, particularly when the variable set is large

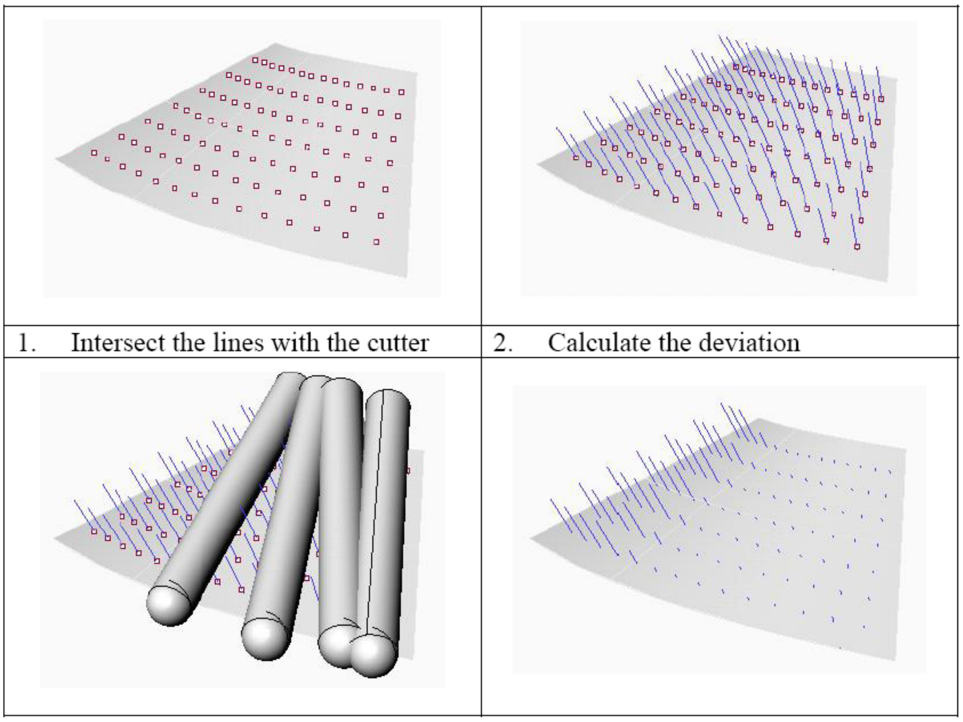


FIGURE 3. Procedure of the machining error estimation with z-buffer method

[17,18]. This is the case in the optimization-based tool path planning. The procedures of SPSA are described as follows:

Step 1: *Initialization*

An initial guess was required to determine the values of the non-negative coefficients a, A, c, α, and γ in the following SPSA gain sequences:

$$\begin{aligned} a_k &= a/(A+k)^{\alpha} \\ c_k &= c/k^{\gamma}, \end{aligned} \tag{2}$$

where k acts as the iteration counter. The values of the coefficients were essential to the performance of SPSA. We chose these values based on the guidelines provided by [15]. A was 10% (or less) of the maximal number of allowable iterations and a was chosen so that the multiplication of a_k and the magnitude of elements in $\hat{g}_0$ $(\hat{\theta}_0)$ were approximately equal to the lower bound of the desired magnitude change among the elements of θ in the early iterations. Moreover, α and γ were set to be 0.602 and 0.101. In the case of obtaining perfect measurements of $E(\theta)$, c must be chosen as a small positive number. Conversely, c is approximately equal to the standard deviation of the measurement noise.

Step 2: *Generation of the simultaneous perturbation vector*

The simultaneous perturbation vector Δ_k was composed of p components, in which p was equal to the number of the optimized variables. Each component was independently generated from a Bernoulli ± 1 distribution with a probability of 1/2 for each ± 1 outcome.

Step 3: *Loss function evaluations*

Given a_k, c_k, and Δ_k, this step was used to obtain two measurements of the deviation based on the simultaneous perturbation around the current $\hat{\theta}_k$: $\mathrm{M}(\hat{\theta}_k + c_k\Delta_k)$ and $\mathrm{M}(\hat{\theta}_k - c_k\Delta_k)$.

Step 4: *Gradient approximation*

Gradient approximation $\hat{g}_k(\hat{\theta}_k)$ was used to estimate the unknown gradient $g(\theta) \equiv \frac{\partial E}{\partial \theta}$ when $\theta = \hat{\theta}_k$. Equation (3) may be used to obtain $\hat{g}_k(\hat{\theta}_k)$, where Δ_{ki} is the ith component of the Δ_k vector.

$$\hat{g}_k(\hat{\theta}_k) = \frac{\mathbf{M}(\hat{\theta}_k + c_k\Delta_k) - \mathbf{M}(\hat{\theta}_k - c_k\Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{3}$$

Step 5: *Updating θ estimation*

Equation (4) is the general recursive SA form used to update $\hat{\theta}_k$ to a new value, $\hat{\theta}_{k+1}$.

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k\hat{g}_k(\hat{\theta}_k) \tag{4}$$

Step 6: *Iteration or termination*

The SPSA procedure iteratively searches for an optimal solution from Step 2 to Step 5. The procedure terminates if the maximal allowable number of iterations has been reached; conversely, we return to Step 2 with $k + 1$ replacing k.

2.4. **Implementation results.** The simulation tests were conducted under various conditions to demonstrate the effectiveness of the proposed method. The following two independent parameters were varied in the tests: (1) the number of cutter locations (or the surface ruling lines), and (2) the maximum number of iterations for terminating the SPSA procedure. As illustrated in Figure 4, the test geometry was a ruled surface that was constructed with two cubic Bézier curves, each defined by four control points. An ϕ-2 ball-end cutter with a 25 mm cutting length was used. The other parameters in the machining error estimation by using the z-buffer method are summarized in Table 1. Various numbers of points were sampled in the u and v directions because the test surface resembled a narrow strip. The Euclidean distance between two interpolated tool positions was chosen as 0.1 mm, that is, the distance between the center points of any two tool positions was smaller than 0.1 mm.

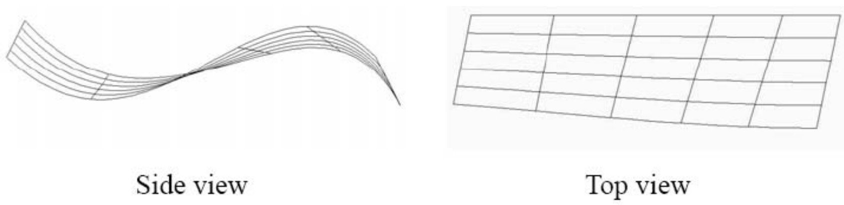


Figure 4. Test surface and its control points

Table 1. Parameters in the machining error estimation by using the z-buffer method

Number of sampling points in u direction	350
Number of sampling points in v direction	10
Minimum Euclidean distance between interpolated tool positions	0.1 mm

The SPSA procedure consisted of three stages that corresponded to various numbers of cutter locations. The initial condition was that all scalars of t and b were equal to zero when the number of ruling numbers was 35. To increase the converge rate of the search process, the optimal solution that was obtained at the first stage formed a part of the initial guess at the second stage when the number of ruling numbers was increased to 70. The initial guess at the newly added 35 cutter locations was chosen as zero. The initial condition in the case of 140 cutter locations was determined in a similar manner, that is, it inherited the optimal solution that was obtained when the number of ruling numbers is 70. The values of a, A, and c were chosen as 0.05, 10, and 0.5, respectively. The simulation results are summarized in Table 2. Two values are illustrated in each cell of the table. The first value represents the average deviation that was obtained from ten independent replications of the same test. This averages the variation due to the stochastic nature of the SPSA method. The second value in the parenthesis indicates the improvement percentage of the error over the result that was generated by a method that is commonly used in practice. In this method, the tool moves along the surface rulings without any optimization or sophisticated strategy.

Table 2. Simulation results of the SPSA-based tool positioning method

			# of cutter location		
			35	70	140
SPSA	# of iterations	100	20.71 (51%)	13.11 (62%)	9.51 (72%)
		200	17.45 (59%)	10.13 (71%)	7.90 (76%)
Practical method			42.14	34.48 (18%)	33.53 (20%)

Several important observations were obtained from the simulation results. First, the proposed method reduced the error by more than 50% compared to that of the common method. Both methods demonstrated favorable results as the number of cutter locations increased. However, the error reduction rate was limited. In the SPSA method, for example, the improvement was 11% by increasing the number of cutter locations from 35 to 70, however it reduced to 5% when the number of cutter location increased from 70 to 140. A similar tendency was observed as the number of iterations increased. In addition, increasing the number of iterations helps to obtain an optimal solution with a longer search process. As illustrated in Figure 5, the solution that corresponded to 140 ruling lines converged when the iteration was approximately 200, however, 35 and 70 ruling lines required a minimum of 300 iterations to converge. A tradeoff may be made between the solution quality and the computational time in practical use. The proposed method is stable under a few conditions. The objective function in the SPSA-based optimization is an approximation to the machining error. The test results confirmed that the error may be effectively reduced in this situation. Another assumption is that the parameter values in SPSA were appropriately selected to ensure the effectiveness of the method.

3. **Comparison with Local Adjustment Methods.**

3.1. **Greedy tool positioning method.** The common method for reducing the machining error in 5-axis flank milling is to locally adjust the tool position and orientation at discrete cutter locations. Most of the previous studies [5,7-11,13] adopted such an approach. The adjustment usually involves the following two parameters of the tool motions: offset and rotation. The offset translates the tool axis along a predefined direction by a specific distance and the rotation rotates the tool orientation with respect to a reference axis. Performing offset and rotation may be either sequential or simultaneous in various studies, and involve univariate or bivariate optimizations, respectively.

Most previous studies applied the tool positioning strategy at individual cutter locations independently. Regardless of optimization or heuristics-based strategy, such methods belong to the category of the greedy approach. This approach selects the locally optimal

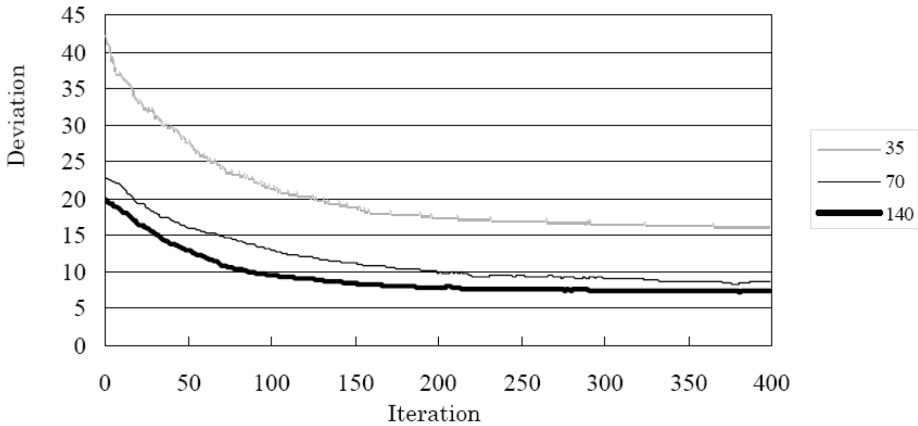


Figure 5. Converge curves with different conditions

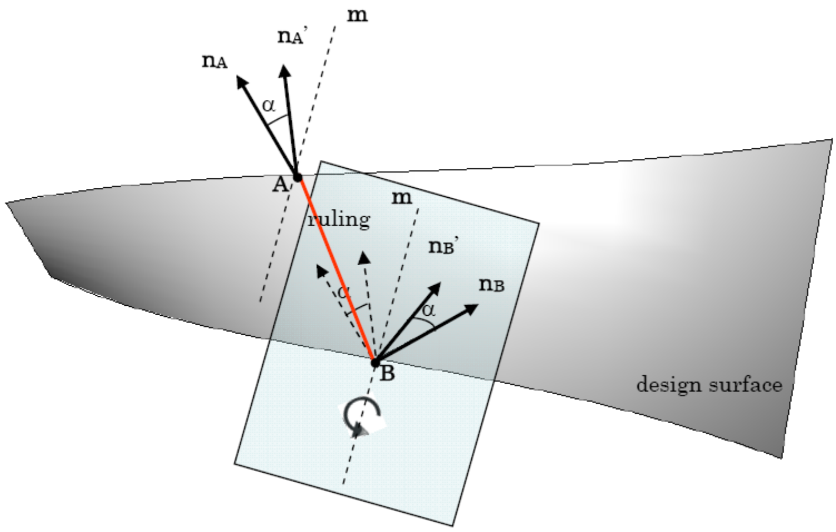


Figure 6. Heuristic search for optimal tool axis

choice at each stage with the aim of determining the global optimum. A greedy algorithm that independently adjusts the cutter locations leads to a suboptimal solution in the case of 5-axis tool path planning [14]. To support this statement, we compared the performance of the SPSA method with a heuristics-based greedy approach that was proposed by the previous study [7]. Another purpose of this comparison was to demonstrate the advantage of our method over the greedy approach in general.

The procedure of the greedy method is described as follows: a set of ruling lines with equal parametric intervals is first generated from the design surface. These ruling lines determine the initial tool orientations. The next step is to calculate the average direction $\mathbf{m}$ between the surface normal vectors ($\mathbf{n}_{\mathrm{A}}$ and $\mathbf{n}_{\mathrm{B}}$) at the end points of each ruling (see Figure 6). The two normal vectors are then rotated about the direction $\mathbf{m}$, producing a set of tool axes. The rotation angle α is limited by an upper bound α^*. A one-dimensional search is subsequently performed to determine the position around the considered ruling line at which the machining error is minimal.

3.2. **Implementation results.** This section presents the comparison result between the greedy method and the SPSA-based method. The aim was to demonstrate that the SPSA-based method usually outperforms the greedy method. For an unbiased comparison, the

TABLE 3. Comparison between the greedy and SPSA methods

	Number of cutter locations		
	35	70	140
Greedy method	33.37 (21%)	14.69 (57%)	11.77 (65%)
SPSA (angle)	32.63 (23%)	11.76 (66%)	9.57 (71%)

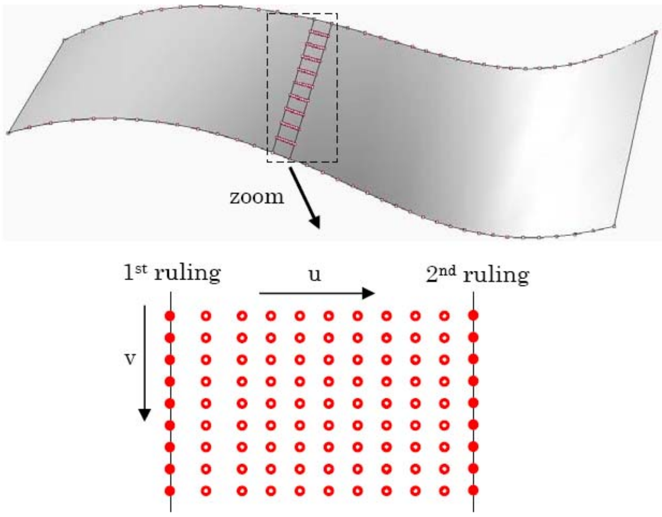


FIGURE 7. Schematic illustration of the error distribution between ruling lines

variable to be perturbed in the SPSA method was changed to the rotation angle α that was described in the previous section. The design surface, the cutter, the gain sequences of SPSA, and the parameters in the machining error estimation remained the same as the condition that was described in Section 2.4. The simulation results are displayed in Table 3. The SPSA method produced a smaller machining error than the local adjustment method in all three cases. This result is based on the fact that, in the optimization field, the sum of the local optima is not equal to (and usually worse than) the global (or near-global) optimum. The search for local optima at discrete stages (the greedy method) results in an inferior result than that of a method that performs a near-global search (the SPSA method). The difference between these methods is more distinct as the number of cutter locations increases. The percentage indicates the improvement of the machining error compared to the result that was generated by following the rulings (see the practical method in Table 3).

The analysis that is illustrated in Tables 4 and 5 supports this argument. They correspond to the test results of the case of 35 ruling lines. The cell values represent the machining error that was estimated at a sampling point on the machining surface. Figure 7 illustrates the surface area from which the sampling points were generated. A total 99 sampling points were observed in this case, as follows: 81 points on the surface area between the adjacent ruling lines and 18 points along the ruling lines. Table 4(a) illustrates the error distribution of the first group of points, which are marked as hollow circles in Figure 7. Table 4(b) illustrates the error distribution along the two ruling lines (solid circles in Figure 7). The results of the SPSA method are displayed in Table 5 in a similar manner. By comparing Table 4(b) and 5(b), the greedy method produces a smaller error along the two ruling lines ($0.0253 < 0.17$ and $0.18 < 0.191$). However, it results in a markedly greater deviation on the in-between area ($1.7937 > 0.6898$ in this case). Consequently, the greedy tool positioning produces a tool path that is not a global optimum.

Table 4. Error distribution produced by the greedy-based method

(a) Deviation on the area between the adjacent ruling lines

	u direction								
v direction	0.0087	0.0130	0.0159	0.0174	0.0176	0.0165	0.0151	0.0124	0.0084
	0.0101	0.0138	0.0161	0.0172	0.0178	0.0172	0.0153	0.0121	0.0076
	0.0115	0.0148	0.0176	0.0190	0.0191	0.0179	0.0154	0.0123	0.0082
	0.0144	0.0179	0.0200	0.0208	0.0203	0.0193	0.0172	0.0138	0.0092
	0.0181	0.0209	0.0228	0.0238	0.0235	0.0220	0.0193	0.0154	0.0102
	0.0218	0.0257	0.0274	0.0277	0.0269	0.0247	0.0218	0.0180	0.0131
	0.0255	0.0310	0.0320	0.0320	0.0312	0.0292	0.0259	0.0215	0.0159
	0.0292	0.0374	0.0384	0.0381	0.0366	0.0339	0.0300	0.0249	0.0192
	0.0330	0.0450	0.0452	0.0442	0.0421	0.0393	0.0353	0.0302	0.0239
Total	1.7937								

(b) Deviation around the ruling lines

	u direction	
	1st ruling line	2nd ruling line
v direction	0.0040	0.0226
	0.0037	0.0220
	0.0034	0.0213
	0.0031	0.0206
	0.0028	0.0200
	0.0025	0.0193
	0.0022	0.0187
	0.0019	0.0180
	0.0016	0.0173
Total	0.0253	0.1800

Figure 8 illustrates the error distributions in 3D contour for both results. The greedy method produces larger machining errors between cutter locations than the SPSA-based method.

Table 6 compares the computational time required by both methods with various numbers of cutter locations. The same test conditions, as illustrated in Table 3, were used. The search process in the SPSA method required a longer period than the greedy method in all three cases. Both methods demonstrate a similar tendency in that the computational time increases in conjunction with the number of cutter locations. The increase is almost linearly proportional to the number.

The comparison of the these results with those in Section 2.4 revealed several insights into the tool path planning based on SPSA. First, the method of perturbing the rotation angle of the tool axis exhibited inferior performance to that of perturbing the scalars of t and n at the end points of the surface rulings. The adjustment of the tool position with those scalars allowed the cutter to move with higher degrees of freedom at each cutter location in 3D. This generated a larger solution space for the SPSA to search for a superior solution. Second, increasing the number of ruling lines (or the number of cutter locations) reduced the machining error, regardless of the planning method. However, the reduction rate was limited, as illustrated in Figure 9. The space cannot be markedly improved when the number of cutter locations is large. The tool paths that were generated by using the three methods are illustrated in Figure 10. The line segments in the figure indicate the tool axis at each cutter location. The number of cutter locations was 70 and the number

TABLE 5. Error distribution produced by the SPSA method

(a) Deviation on the area between the adjacent ruling lines

	u direction								
v direction	0.0032	0.0037	0.0041	0.0042	0.0041	0.0041	0.0040	0.0040	0.0041
	0.0053	0.0062	0.0069	0.0074	0.0075	0.0076	0.0078	0.0080	0.0083
	0.0068	0.0080	0.0089	0.0096	0.0099	0.0102	0.0105	0.0109	0.0114
	0.0077	0.0090	0.0101	0.0110	0.0113	0.0117	0.0122	0.0127	0.0133
	0.0079	0.0093	0.0105	0.0114	0.0118	0.0122	0.0127	0.0133	0.0140
	0.0074	0.0088	0.0100	0.0110	0.0113	0.0117	0.0122	0.0129	0.0136
	0.0062	0.0076	0.0087	0.0096	0.0099	0.0102	0.0107	0.0112	0.0119
	0.0044	0.0057	0.0066	0.0073	0.0075	0.0077	0.0080	0.0085	0.0090
	0.0026	0.0030	0.0037	0.0042	0.0041	0.0042	0.0043	0.0046	0.0050
Total	0.6898								

(b) Deviation on the adjacent ruling lines

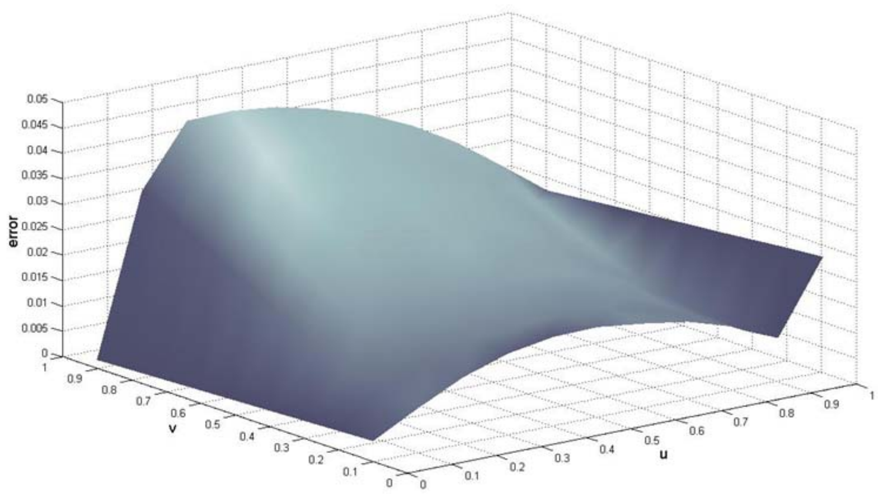
	u direction	
v direction	0.0030	0.0220
	0.0100	0.0250
	0.0160	0.0290
	0.0190	0.0300
	0.0210	0.0280
	0.0230	0.0250
	0.0240	0.0190
	0.0260	0.0110
	0.0280	0.0020
Total	0.1700	0.1910

TABLE 6. The computational time required under various conditions

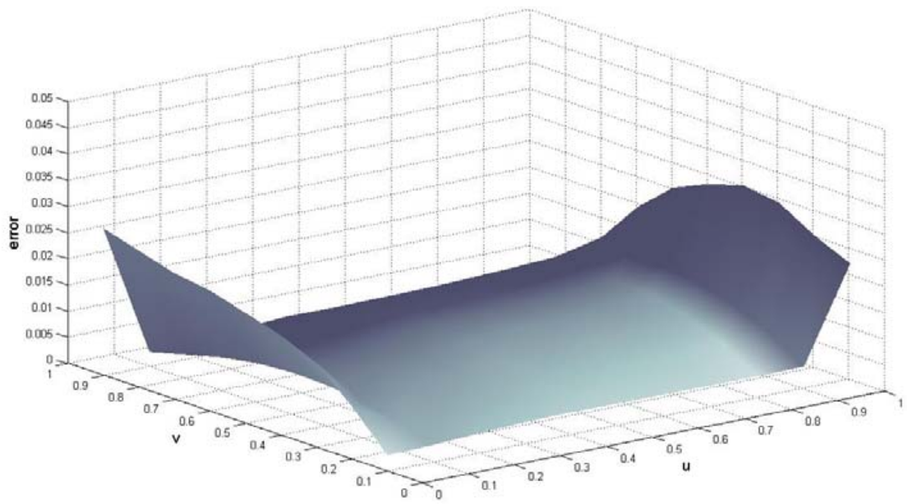
	Number of cutter locations		
	35	70	140
Greedy method	72.6 (s)	150.6 (s)	321.4 (s)
SPSA (angle)	112.2 (s)	256.7 (s)	514.5 (s)

of iterations in the two SPSA methods was 200. It was observed that perturbation of the rotation angle produced a smoother tool pose change as compared to the other two methods.

4. **Conclusion.** Machining error control is a challenging task in the 5-axis flank milling of a ruled surface. The following two approaches were proposed by previous studies: heuristics and optimization-based methods. These methods form part of the greedy approaches, which locally adjust the tool position at distinct cutter locations. The tool path that is produced in this manner is not the optimal result in terms of the total error on the machined surface. The deviation in the surface region between cutter locations may deteriorate to a greater degree; however, the error near the cutter locations is reduced. A holistic approach that adjusts the tool position at all cutter locations simultaneously is required to overcome this problem. Conversely, the gradient-based optimization methods are not applicable to the problem because a closed-form representation of the machining error estimation does not exist. Therefore, we presented a tool path planning method based on simultaneous perturbation stochastic approximation. All of the cutter locations



(a) Error distribution of the greedy method



(b) Error distribution of the SPSA method

Figure 8. Comparison of machining error distributions produced by the greedy and SPSA methods

of a tool path were optimized simultaneously, thus overcoming the deficiency in the previous greedy methods. The SPSA is particularly useful when the number of variables to be optimized is large. The simulation tests demonstrated that the SPSA-based tool positioning was superior to local adjustment methods by producing smaller machining errors. The difference was diminished with the increase of the number of cutter locations. Moreover, perturbing the tool axis along the tangent and bi-normal directions of the surface rulings produces a superior result than by perturbing a single rotation angle of the tool axis. This study provides a simple, systematic, and effective method for improving machining quality in 5-axis flank milling of ruled surfaces.

The percentage of the error reduction by the SPSA method, compared with the other methods, increases in conjunction with the number of cutter locations. The computational time required by the tool path planning also increased. This may limit the practical value of the method. A potential method of solving this problem is to combine SPSA with

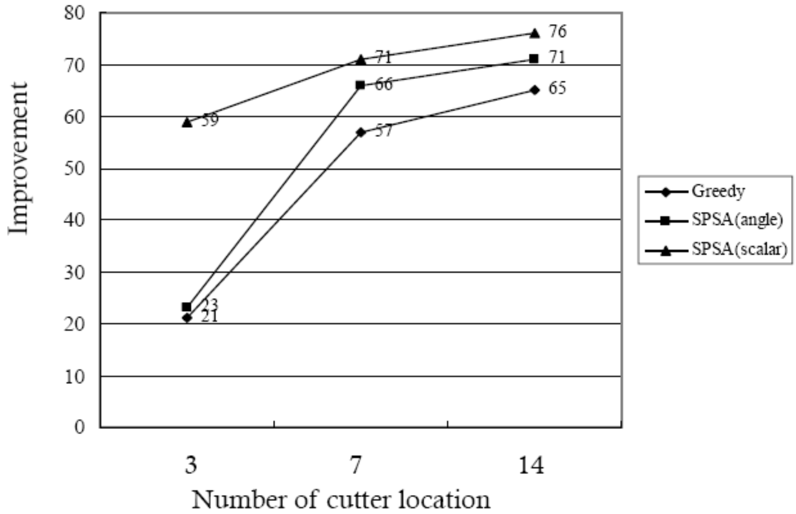
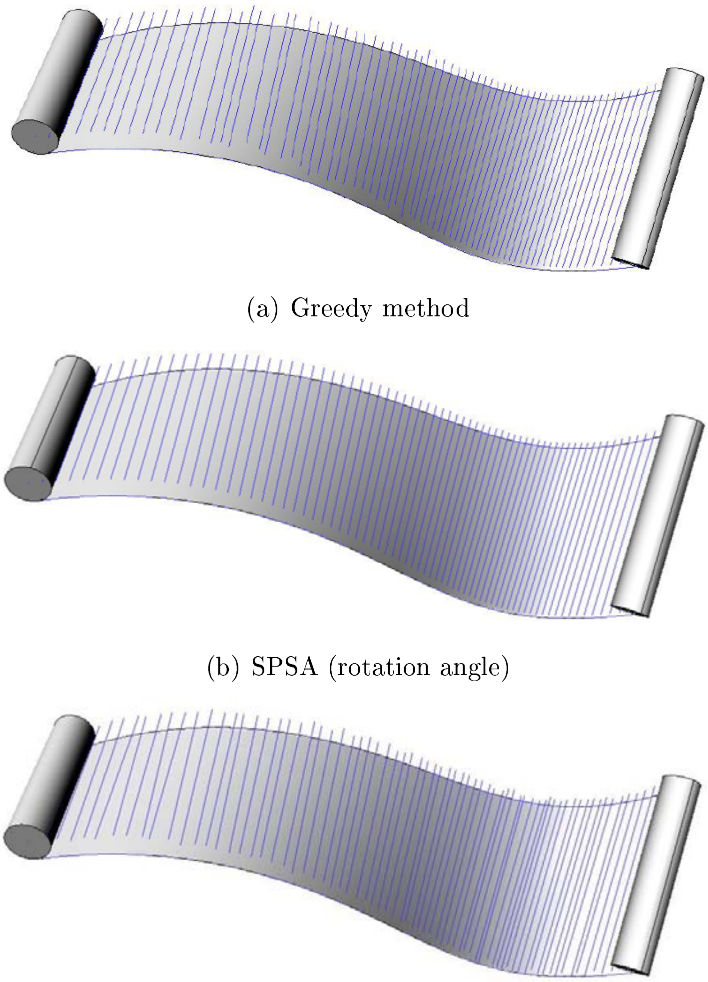


Figure 9. Error reduction rate versus the number of cutter locations



(a) Greedy method

(b) SPSA (rotation angle)

(c) SPSA (tangent and bi-normal)

Figure 10. Tool paths generated from three different methods

evolutionary computing techniques. The SPSA helps to generate favorable trial solutions in one iteration and an evolutionary computing algorithm (for example, GA or PSO) guides the search process. In addition, GPU (Graphics Processing Units)-based parallel processing provides a simple but powerful method to accelerate the optimization process involved in the tool path planning. Each processing unit is responsible for the error estimation of one cutter location. Future studies will integrate parallel processing techniques to enhance the computation efficiency in the current method. Another potential extension is to use machining error as a constraint in the optimization while minimizing other objectives, such as energy consumption or cutting forces.

REFERENCES

[1] C. H. Chu and J. T. Chen, Automatic tool path generation for 5-axis flank milling based on developable surface approximation, *The International Journal of Advanced Manufacturing Technology*, vol.29, no.7-8, pp.707-713, 2006.
[2] G. Stute, A. Storr and W. Sielaff, NC programming of ruled surfaces for five axis machining, *Annals of the CIRP*, vol.28, no.1, pp.267-271, 1979.
[3] X. W. Liu, Five-axis NC cylindrical milling of sculptured surfaces, *Computer-Aided Design*, vol.27, no.12, pp.887-894, 1995.
[4] E. L. J. Bohez, S. D. R. Senadhera, K. Pole, J. R. Duflou and T. Tar, A geometric modeling and five-axis machining algorithm for centrifugal impellers, *Journal of Manufacturing Systems*, vol.16, no.6, pp.422-436, 1997.
[5] J. M. Redonnet, W. Rubio and G. Dessein, Side milling of ruled surfaces – Optimum positioning of the milling cutter and calculation of interference, *The International Journal of Advanced Manufacturing Technology*, vol.14, no.7, pp.459-465, 1998.
[6] J. J. Lee and S. H. Suh, Interference-free tool-path planning for flank milling of twisted ruled surfaces, *The International Journal of Advanced Manufacturing Technology*, vol.14, no.11, pp.795-805, 1998.
[7] D. M. Tsay and M. J. Her, Accurate 5-axis machining of twisted ruled surfaces, *Journal of Manufacturing Science and Engineering*, vol.123, pp.731-738, 2001.
[8] S. Bedi, S. Mann and C. Menzel, Flank milling with flat end cutter, *Computer-Aided Design*, vol.35, pp.293-300, 2003.
[9] C. Menzel, S. Bedi and S. Mann, Triple tangent flank milling of ruled surfaces, *Computer-Aided Design*, vol.36, pp.289-296, 2004.
[10] J. Senatore, F. Monies, J. M. Redonnet and W. Rubio, Improved positioning for side milling of ruled surfaces: Analysis of the rotation axis's influence on machining error, *International Journal of Machine Tools & Manufacture*, vol.47, pp.934-945, 2007.
[11] C. Lartigue, E. Duc and A. Affouard, Tool path deformation in 5-axis flank milling using envelope surface, *Computer-Aided Design*, vol.35, pp.375-382, 2003.
[12] C. J. Chiou, Accurate tool position for five-axis ruled surface machining, *Computer-Aided Design*, vol.36, pp.967-974, 2004.
[13] H. Gong, L. X. Cao and J. Liu, Improved positioning of cylindrical cutter for flank milling ruled surfaces, *Computer-Aided Design*, vol.37, pp.1205-1213, 2005.
[14] C. H. Chu, P. H. Wu and W. T. Lei, Tool path planning for 5-axis flank milling of ruled surfaces considering CNC linear interpolation, *Journal of Intelligent Manufacturing*, 2010.
[15] J. C. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, Wiley, Hoboken, NJ, 2003.
[16] P. H. Wu, Y. W. Li and C. H. Chu, Optimized tool path generation based on dynamic programming for five-axis flank milling of rule surface, *International Journal of Machine Tools & Manufacture*, vol.48, pp.1224-1233, 2008.
[17] S. Tan, X. Cheng and H. Xu, An efficient global optimization approach for rough set based dimensionality reduction, *International Journal of Innovative Computing, Information and Control*, vol.3, no.3, pp.725-736, 2007.
[18] B.-K. Bao, H. Xi, B. Yin and Q. Ling, Two time-scale gradient approximation algorithm for adaptive Markov reward processes, *International Journal of Innovative Computing, Information and Control*, vol.6, no.2, pp.655-666, 2010.