Optimizing of NC Tool Paths for Five-Axis Milling using Evolutionary Algorithms on Wavelets

Klaus Weinert
University Dortmund, Dep. of Machining Technology
Baroperstr. 301
D-44227 Dortmund
(+49) 231 755-2783
weinert@isf.de

Andreas Zabel
Univ. Dortmund, Dept. of Machining Technology
Baroperstr. 301
D-44227 Dortmund
(+49) 231 755-2708
zabel@isf.de

Heinrich Müller
Univ. Dortmund, Dept. of Computer Graphics
Otto-Hahn-Str. 16
D-44221 Dortmund
(+49) 231 755-6324
mueller@ls7.informatik.uni-dortmund.de

Petra Kersting
Univ. Dortmund, Dept. of Machining Technology
Baroperstr. 301
D-44227 Dortmund
(+49) 231 755-2113
pkersting@isf.de

ABSTRACT
Computer aided NC-path generation of five-axis milling using a standard CAM-system does usually not take machine dynamics and kinematics into account. This results in machine movements which are often not smooth enough and lead to a deficient surface quality. In order to reduce undesirable abrupt motion changes, an approach for optimizing the NC-path by using a standard evolution strategy is shown in this paper as well as first results of applying this algorithm to the five-axis milling process.

Track: Real-World Applications

Categories and Subject Descriptors
I.6 [Simulation and Modeling]: Model Development

General Terms
Algorithms

Keywords
Evolution Strategy, Wavelets, Five-Axis Milling, Mechanical Engineering, Application

1. INTRODUCTION
Compared to three-axis milling, the five-axis process offers further possibilities of cutter movements [20]. Additional to the moving perpendicular to the three coordinate-axes, cycling around the axes is also possible (Figure 1a). These additional degrees of freedom allow for example the use of shorter milling tools (Figure 1b and Figure 1c) which are usually much stiffer, show less tendency of deflection and lead to an improved surface quality [14]. Furthermore, the possibility of using different angle values allows the milling in one workpiece clamping and reduces the number of steps in the process chain.

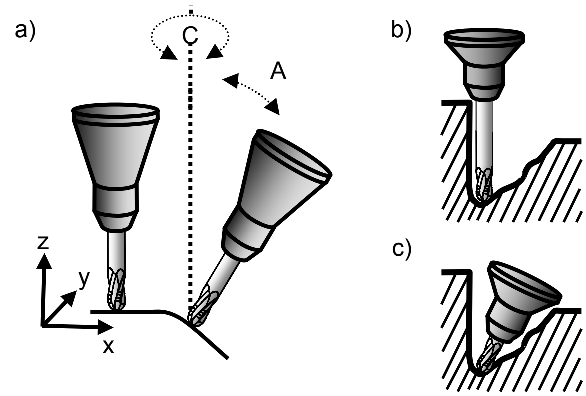

Figure 1. a) Illustration of five-axes milling. b) Example of three- and c) of five-axes milling.

Modern machine tools are usually CNC (Computer Numerical Control) driven and the applicable NC-paths are generated by CAD/CAM-systems based on the target geometry of the desired workpiece [20]. The special machine kinematics, however, are usually not considered, although they are important issues for the quality of the resulting workpiece surface. In particular, jerky motion changes of heavy machine components could cause surface defects on the workpiece (see Figure 2). This observation motivates to analyze the motion path of individual kinematic elements, and to optimize the NC-path in order to improve surface structures.

For that purpose, an algorithm has been developed and integrated in a simulation environment for five-axis milling which is placed in the process chain between the CAM-system and the real milling process. The main idea is to analyze the input NC-paths during the virtual milling process and to optimize the roll and tilt angles along the milling path while leaving the locations of intervention of the tool unchanged. This path optimization is based on an evolutionary algorithm (EA) which works on a wavelet transformation of the mentioned angles and optimizes the coefficients of the wavelet functions. This approach is carried out under the constraint that the generated tool path should be technological well suited, free of collisions and leads to smooth and jerk free-paths which improves the surface quality for a given workpiece.

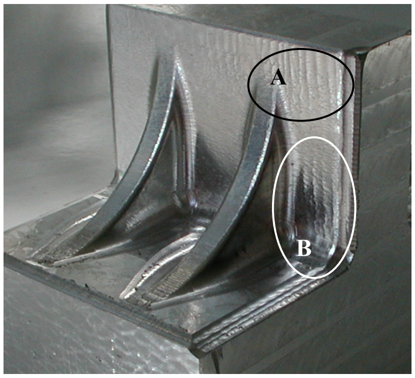

Figure 2. Real workpiece. A/B: Surface defects resulting from vibrations of milling machine elements.

NC-path optimization is a relevant topic of interest and several approaches with different optimization objectives have been published in the past. Castelino at. el. [2] for example describe an algorithm for minimizing the non-productive tool time, Yoon [22] evaluates optimal cutting positions with respect to the avoidance of tool tip gouging and My at el. [10] have developed an approach for five-axis tool path optimization by clustering the optimal cutting direction vector field. Stautner and Zabel [14] introduced a prototype of a semi-automatic optimization system which uses evolutionary algorithms to optimize multi-axis tool paths and to convert given three-axis paths into optimized multi-axis ones with regard to cutting force reduction and collision prevention. Surmann, Kalveram and Weinert [15] describe a simulation system which predicts the vibration behavior of the cutting tool along arbitrary NC-programs for the milling of sculptured surfaces. This software allows the prediction of those segments of an NC-program which are sensitive to chatter vibrations. Their objective is the adaptation of the process parameters using this knowledge. Analysis of research literature shows that the influence of machine kinematics has not been taken into account by now.

In the next section, the main principles of the wavelet transform are briefly reviewed. The optimization algorithm is described in Section 3. Results of a simulation-based evaluation are presented in Section 4. Finally, the approach is summarized.

2. WAVELET TRANSFORM

The following brief introduction is limited to those aspects of the wavelet transform which are relevant for this paper. Further information can be found in literature [11, 12, 21].

Like the classical Fourier theory [5], the wavelet transform represents a given function as a linear combination of a set of basis functions. In contrast to the sine and cosine functions used by the Fourier transform, the wavelet functions are usually equal to 0 outside a finite interval. In this way, the wavelet transform is canonically suited for local analysis of restricted regions of interest.

The basis functions of a wavelet transform are distinguished in the scaling functions and the wavelet functions (wavelets for short). The scaling function is used to represent a crude approximation of the function while the wavelets represent details of different resolution or frequency. To simplify matters, the scaling functions are not further addressed in the following descriptions.

The wavelets $\psi_{j,k}$ are obtained from a basis – the so-called *mother wavelet* ψ – by dilatation and translation,

$$\psi_{j,k}(t) = \sqrt{2^j}\,\psi(2^j t - k),$$

with $(j,k) \in \mathbf{Z}^2$. Figure 3 shows two common examples of mother wavelets.

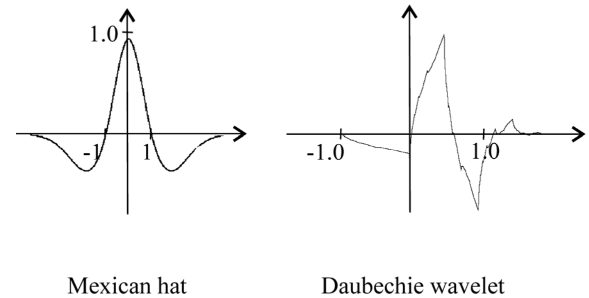

Figure 3. Examples of mother wavelets.

Since the functions $f(t)$ to be optimized are represented in a discrete way, that is by sample values at discrete locations $t=0,\ldots,n-1$, the *discrete wavelet transform* is required:

$$f(t) = c_{0,0}\phi_{0,0}(t) + \sum_{j=0}^{m-1}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(t),$$

where $n=2^m$, $\phi_{0,0}$ is a *scaling function*, $c_{0,0} \in \mathbb{R}$ is a *scaling coefficient*, and $d_{j,k} \in \mathbb{R}$ are the *wavelet coefficients*. n input values result in n (scaling and wavelet) coefficients after the transformation.

The explicit function $f(t)$ can be recalculated from the wavelet coefficients by the inverse wavelet transform. Efficient algorithms exist for both directions. In contrast to the complexity of the Fourier transform $O(n \log(n))$, the complexity of the Wavelet transform is $O(n)$ [11, 12, 21].

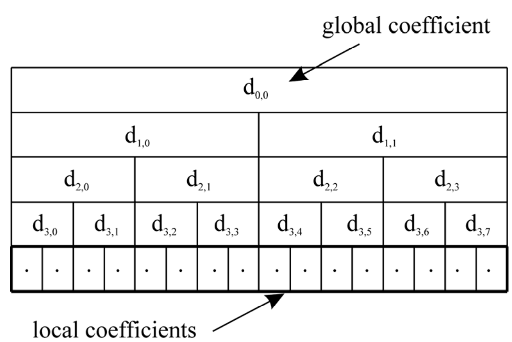

Figure 4. Arrangement of the wavelet coefficients. The horizontal axis represents the domain of the given function, the vertical axis the level of detail or frequency.

Accentuating the influence of the coefficients and their corresponding wavelets on the input function $f(t)$, an intuitive graphical arrangement of the coefficients is used. Figure 4 shows an example of a function existing of 32 data values; in real applications several thousand values are usual. The horizontal axis represents the domain of the given function, the vertical axis the level of detail or frequency. This means that the first row contains the coefficient belonging to the function with the largest dilatation, the last row comprises the coefficients corresponding to fine details. The coefficient related to the function of largest dilatation has influence on the whole input function, whereas the other wavelet coefficients affect just intervals of decreasing length.

3. OPTIMIZING TOOL PATHS

The approach of improving a given NC-path is realized in a simulation environment of the milling process. In order to optimize the process parameters, a high amount of parameter combinations have to be evaluated. Therefore, a fast and realistic computer simulation is required. For that purpose, the relevant process characteristics have to be represented as detailed and realistic as possible. The precision and running time of the simulation is basically influenced by the choice of the model for the workpiece and milling tool. Modeling a five-axis milling process, several requirements have to be regarded, especially the support of undercuts in tolerable precision and running time. Analysis of research literature shows different modeling approaches [15, 19]. A satisfying solution is introduced by [17] which depends on a special discrete workpiece representation (dexel field) and a CSG (Constructive Solid Geometry [4])-based model for the milling tool.

This simulation environment is used in order to record the tilt and roll angle as well as the motion curves of selected critical machine elements during the simulation of a given milling machine for a given NC-path. The basic idea of the algorithm presented in the following is to smooth the angle functions with restriction of maintaining its principal characteristics and the limitation of considering the quality of the motion curves of a machine element. For that purpose, three general principles are combined: the wavelet transform in order to achieve a representation of the angle functions suitable to optimization, curve features to express the objectives, and evolutionary strategies in order to optimize the given function. Due to the complexity of the correlation between the variable components, the use of evolutionary algorithms is reasonable in this context. It is evident that the high number of adjustable coefficients and therewith the high number of degrees of freedom is critical for the application of evolution strategies. In order to handle this problem a matrix splitting approach has been developed. A schematic overview of the algorithm is shown in Figure 5 and a more detailed description is given in the following.

3.1 Modification of Wavelet Coefficients

The main idea of improving the curve functions is to transform the angle values into a wavelet representation, to adjust the corresponding wavelet coefficients, and afterwards, to apply the inverse transform. The wavelet transform is elected in this case to achieve a suitable representation for the problem of tool path optimization because this transform offers the possibility of analyzing and modifying only restricted regions of interests. As mentioned in Section 2, this possibility is based on the characteristic that wavelet functions have finite support, that is they are equal to 0 outside a finite interval.

For the purpose of coefficient modification, a classical evolution strategy is used [1, 13]. The basic idea is to work on a population of potential solutions, called individuals, and to manipulate these individuals with evolutionary operators [7].

Each individual consists of a complete set of wavelet coefficients. The individuals are initialized with the coefficients belonging to the input functions. In order to avoid getting curves with worse progression the $(\mu + \lambda)$-strategy is applied. In executed experiments, respectable solutions can be reached with a *(10,5)*-strategy (see Section 4).

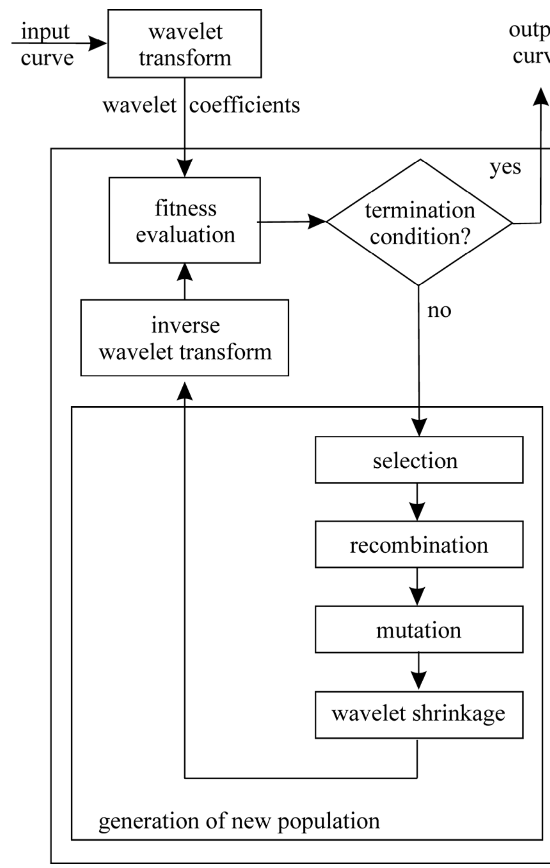

Figure 5. Algorithm overview. The incoming curves are transformed into a wavelet representation and optimized using a classical evolutionary algorithm.

After execution of a sequence of mutations and recombinations, an additional modification known as *wavelet shrinkage* [21] is used which resets all coefficients smaller than a given threshold value to 0. Depending on the magnitude of the threshold, the functions get more or less smoothed. This procedure is applied in order to reduce oscillations of the curves illustrated by an example in Figure 6.

Because it is more convenient to evaluate angle functions in explicit than in wavelet representation, an inverse wavelet transform is applied to every individual. The results are functions of new angles. In order to evaluate the fitness of the individuals, new motion curves are generated by simulating the machine kinematics for the NC paths induced by the new angle values.

Afterwards, the weighting function is evaluated for each individual. For this purpose, the characteristics of the angle functions are considered as well as the ones of the motion curves.

The best μ individuals are chosen for creating the next generation's population (see Section 3.2).

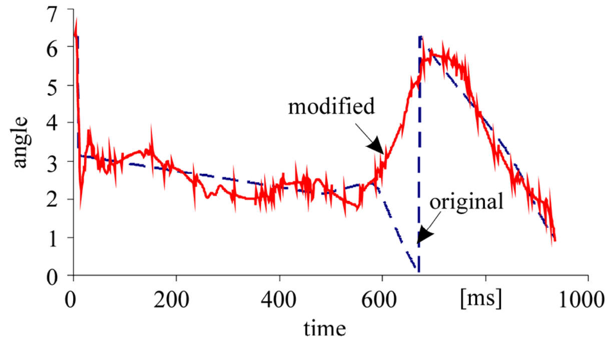

Figure 6. Example of undesired curve oscillations. The dashed lines represent the original data values, the continuous curve shows the modified data.

3.2 Fitness Function Design

Several features of functions seen as curves are used for quantifying the optimality of a solution (individual): the curvature, the length, and the deviation of the modified curve from the original one before modification.

3.2.1 Curvature

Due to the fact that sharp bends in a motion curve result in strong motion changes, the curvature is chosen as important criterion of curve behavior. Hence, curves of small curvature are preferred.

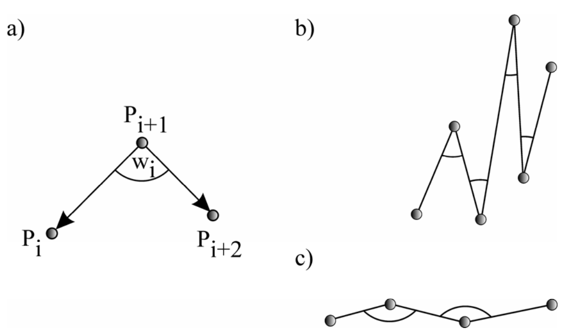

Figure 7. Curvature. a) The curvature is estimated by the sum of absolute cosine values of the angle w_i between two curve segments P_iP_{i+1} and $P_{i+2}P_{i+1}$. b) A curve with high curvature. c) A curve with small curvature.

Evaluating the curvature, the sum of absolute cosines of the angles w_i between two adjacent curve segments P_iP_{i+1} and $P_{i+1}P_{i+2}$ are used (Figure 7a):

$$\kappa = \sum_{i=1}^{n} |\cos w_i| = \sum_{i=1}^{n} \frac{< P_i - P_{i+1}, P_{i+2} - P_{i+1} >}{|P_i - P_{i+1}||P_{i+2} - P_{i+1}|},$$

where the P_i are the angle values at the i-th simulation step.

3.2.2 Curve Length

In order to avoid wavy curve results (compare Figure 8a and Figure 8b), the curve length is chosen as another important feature which is calculated as average sum of Euclidean distances d:

$$l = \frac{1}{n} \sum_{i=1}^{n} d(P_i, P_{i+1}).$$

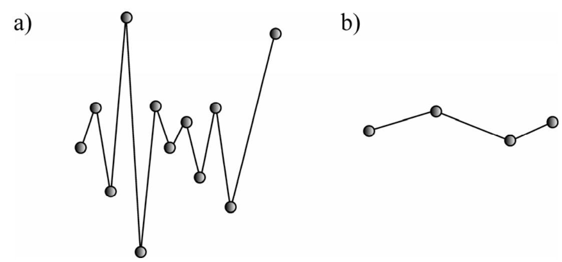

Figure 8. Curve length is another important criteria. The curve represented in b) is shorter and smoother than the curve in a)

3.2.3 Deviation from the Original Curve

Taking just curvature and curve length as features into account, the optimization would always lead to a straight line – independently of the shape of the original curve. This effect is not desired because the angles would not change during the process, and thus, two of the five available degrees of freedom (three along the coordinate axes and two angles) are not used. In order to prevent this effect, the deviation from the original curve is chosen as another important criterion which is evaluated as squared Euclidean distances between the points of the original curve P_i^{orig} and the ones of the modified curve P_i^{mod}:

$$a = \frac{1}{n} \sum_{i=1}^{n} d(P_i^{orig}, P_i^{mod})^2.$$

3.2.4 Complete Fitness Function

The different features of a curve (curvature κ, curve length l and deviation a from the original curve) are provided with individual weights g and are aggregated to the complete evaluation e:

$$e = g^{\kappa}\kappa + g^{l}l + g^{a}a,$$

where $g^{\kappa} \in$ □ is the weight of curvature κ, $g^{l} \in$ □ the weight of curve length, and $g^{a} \in$ □ the weight of deviation.

The complete fitness function is calculated as weighted sum of the evaluation of the tool angles themselves and furthermore of the evaluation of the motion curves of selected critical machine components:

$$q = \sum_{j=1}^{m} g_j (g^{\kappa}\kappa_j + g^{l}l_j + g^{a}a_j),$$

where the index j indicates the curve and $g_j \in$ □ is the weight of the j-th curve.

3.3 Matrix Splitting

The high number of degrees of freedom caused by the high number of wavelet coefficients is critical for the application of evolution strategies. Hence, a technique has been developed to subdivide this difficulty in smaller optimization problems. Therefore, the linear characteristic of the wavelet transform – a function can be represented as a weighted sum of scaling and wavelet functions (Section 2) – is used. The idea is to take different terms out of the sum in order to optimize each of these terms individually using an evolutionary algorithm.

For that purpose, the observation is used that sharp bends in a curve are reflected in large wavelet coefficient values. This property allows smoothing the curve without modifying any of the values, but just the large ones and those in the neighborhood of large coefficients. For the following explanation the kind of matrix notation of the wavelet coefficients introduced in Section 2 is used.

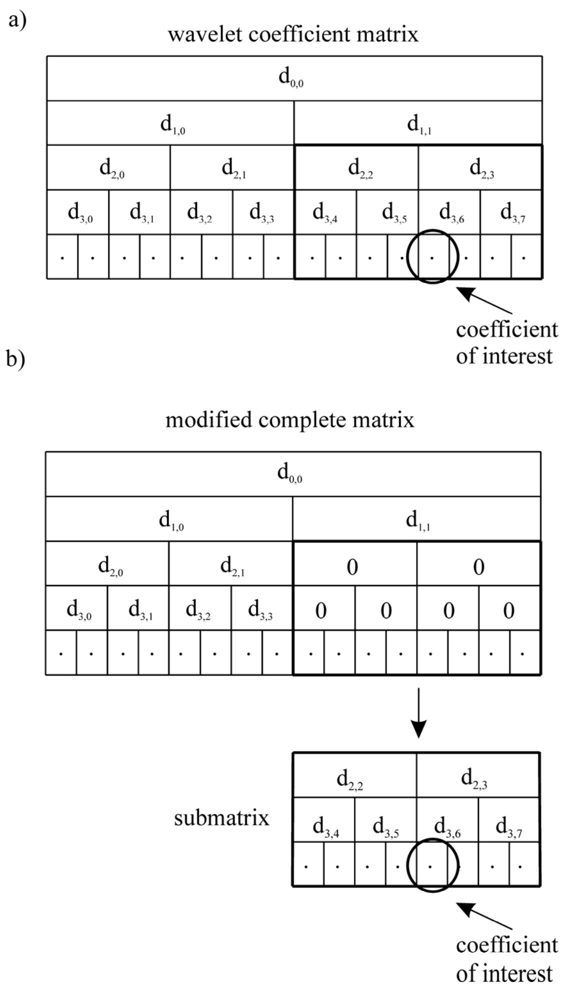

Figure 9. Submatrix selection. a) Generating submatrices for every interesting coefficient. b) Extracting the submatrix by resetting its coefficient values to 0.

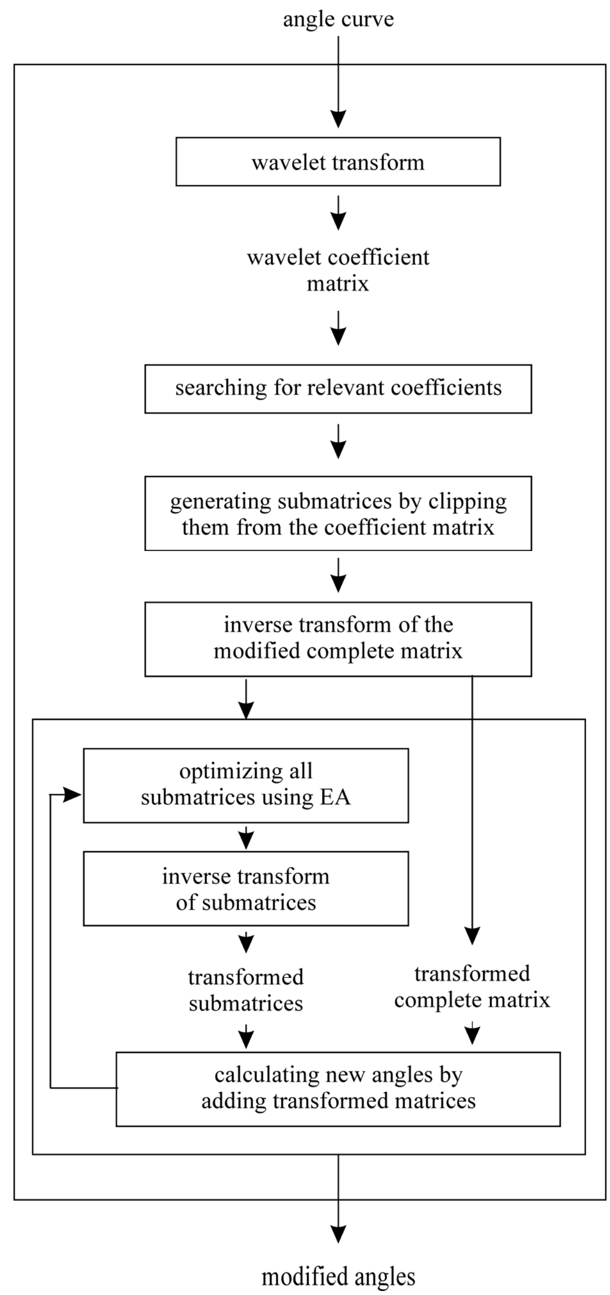

Figure 10. Schematic overview of splitting the matrix in several submatrices and optimizing the different partitions separately.

Scanning the matrix for interesting (large) coefficient values in the bottom row and creating submatrices of constant size consisting of 14 coefficients (Figure 9a) are the basic ideas of the splitting-approach. The submatrices are extracted from the complete structure by resetting its coefficients to 0 (Figure 9b). Afterwards, the inverse wavelet transform is just applied to the modified complete matrix (Figure 10). Each of the small submatrices are optimized separately by a standard evolution strategy before they are also inversely transformed. Due to the linearity of the wavelet transform [8] the complete result is obtained by adding the separate inverse transformed modified matrix and the transformed submatrices. Apparently, performing the inverse wavelet transform on individuals of smaller sizes improves the performance of the algorithm. A short schematic overview of this process is given in Figure 10.

3.4 Collision Handling

After modifying the tilt and roll angles, the NC path has to be newly generated and furthermore, a collision test with respect to the optimized angles is necessary. For that purpose, the complete process is simulated again and detected collisions are eliminated.

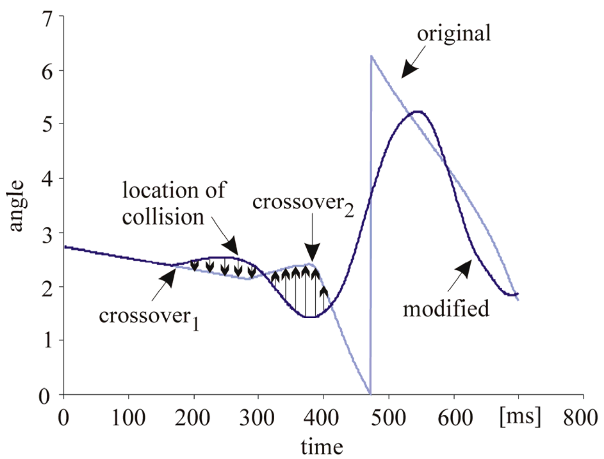

Figure 11. Example of detected collision.

In literature, several algorithms and libraries for collision detection are known. Lin and Gottschalk [8] give a detailed overview over different approaches. In the case of collision detecting between kinematic elements, the library VCollide developed by Hudson et al. [6] was chosen which detects collisions between triangled objects using a fast hierarchical approach. Detecting collisions between the milling tool and the workpiece, an approach adjusting the special workpiece and tool representation was used which is introduced by Weinert at el. [16, 18].

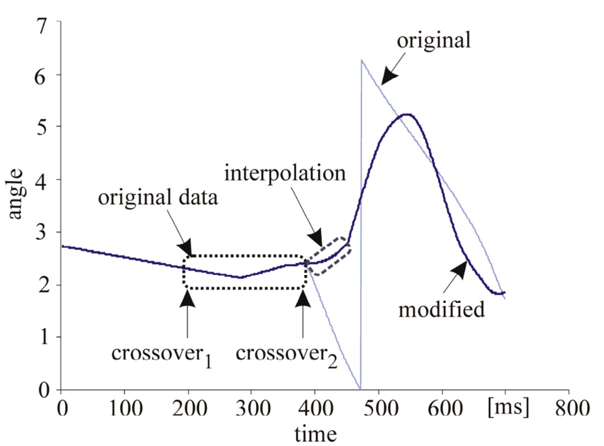

Figure 12. Illustration of collision handling.

Due to the assumption that the original NC-path causes no incidents, the collisions detected during the simulation of the new generated angles can be treated by using the original data in the environment of critical locations. In order to illustrate the procedure of collision handling, an example is shown in Figure 11 and Figure 12. The light-colored curve indicates the input angles, the dark-colored line the modified ones. At the marked time location in Figure 11, a collision is detected. The main idea is to define two points ($crossover_1$ and $crossover_2$ in Figure 11) and to use the original data between these two points.

In order to get a smooth fading from the modified to the original curve (Figure 12), a kind of Hermite interpolation is used [9]. The process of simulating, collision detection and correction has to be repeated as long as the NC-path is modified in order to get legal and collision free tool paths.

4. FIRST EXPERIMENTAL RESULTS

The proposed algorithm was integrated into an existing milling simulation software tool. The simulation was implemented and tested on a Windows XP based, standard 1.6 GHz PC. As programming language C++ was used.

In order to validate the proposal, two examples will be discussed in the following. At first, a small example is shown in Figure 13. For that purpose, the appendant NC-path has been simulated on the parallel kinematic machine Mikromat 6X Hexa [3] and the required data values have been recorded. The continuous curve in Figure 13 represents the original data values, the dashed line the modified ones.

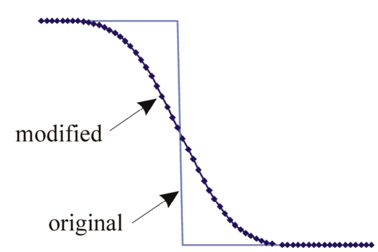

Figure 13. Small example of optimizing an angle curve. The continuous curve represents the original data values, the dashed line the modified ones.

In Figure 14, a second example is shown. Therefore, the machining of a real test workpiece (see Figure 2) has also been simulated on the parallel kinematic machine. The recorded angle curve is shown in Figure 14 and – because of motion sequence recurring – only an excerpt of this curve was considered.

The proposed algorithm has been applied to this extraction with the adjustments shown in Table 1. As termination condition, the number of cycles is restricted. The weights were chosen in a ratio of *5:1:3*. Additionally to the angle curves, only the motion data of the center of the spindle was recorded in the given example. For the according weights g_i a ratio of *3:1* was chosen.

These introduced adjustments of the algorithm parameters have been detected in different experiments. The algorithm is very sensitive to the choice of parameter values which depends primary on the characteristic of the problem. One possibility for detecting suitable parameter values is to put an evolutionary algorithm around the problem in order to find the right parameters.

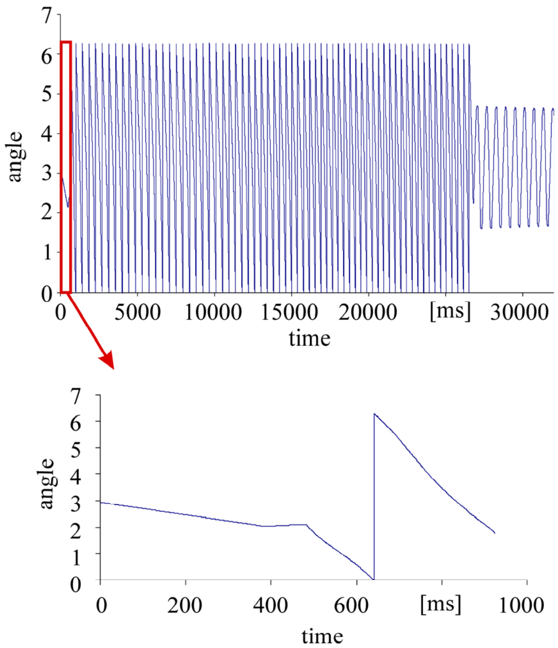

Figure 14: Example of a recorded angle curve. Because the angle sequences recur (top) only an extract of this curve is considered (bottom).

The result is shown in Figure 15. The dashed line represents the original data values, the continuous curve the modified one. Obviously, the sharp bends of the original curve are smoothed, whereas the deviation between the modified curve and the original one is relatively small.

Table 1. Algorithm adjustments

Number of children	λ	10
Number of selected children	μ	5
Max. number of generations	t_{max}	80
Deviation from original curve	a	4
Weights (in ratio of length : curvature : deviation)	$g^l:g^\kappa:g^a$	5 : 1: 3

The data size of the introduced examples is relatively small. In real applications, several thousand values are in use. For that purpose, it is – especially with respect to the process time – not advisable to optimize the whole NC-path at once. The computing time depends in particular on the number of data values, on the number of interesting and chosen coefficients, and on the number of different individuals. In order to accelerate the process and to save memory it is reasonable to optimize data values in smaller groups. Different experiments have shown that a group size about 2^9- 2^{10} values is satisfying. In the given example (Figure 15), the computation time using the matrix splitting approach for a group of about 1000 values is about 180 seconds. Due to the fact, that the introduced evolutionary approach sometimes needs a few runs until a satisfying result is achieved, subdividing the problem into smaller optimization groups is advisable.

The choice of the level of target smoothness depends on the kinematic behavior of the real milling machine for which the NC-path is optimized. Because the accuracy of a standard milling machine is normally much smaller than the computationally accuracy, several experiments are necessary in order to find an expedient correlation between the smoothness achieved by the optimizing algorithm and the mechanical effects, particularly the improvement of the surface quality.

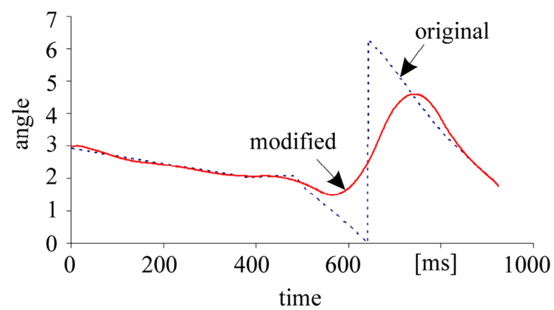

Figure 15: Experimental result. The dashed line represents the original angle data values, the continuous curve the modified ones.

Finally, the results shown in Figure 13 and Figure 14 are discussed with respect to the mechanical advantages. Overall, three main technical advantages can be mentioned. Sudden changes of the tool as well as of other moving parts of the milling machine cause surface defects (see Figure 2). Hence, the surface quality is improved by minimizing the acceleration and the vibrations of the machine components. Furthermore, saving machining time is a second important feature. Because the machining process has to wait until the driving axle has finished its movement, large angle changes from one step to another have a time consuming effect. By tilting the milling tool as well as the other moving machine parts slowly during the milling process, and not completely at once, milling time can be saved. These slow changes are achieved by a smooth curve behavior. The third advantage is the decreasing of machine wear by reduction strong swivel movements.

5. CONCLUSION

In this paper an approach has been presented in order to find improved tilt and roll angles for NC tool paths which reduce the vibrations of moving machine components by minimizing sharp bends in motion curves. For that purpose, methods of the wavelet theory and evolutionary algorithms have been combined. First simulation tests on real NC data show the successful applicability of this combination.

6. ACKNOWLEDGMENTS

This research was supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center.

7. REFERENCES

[1] Beyer, H.-G. and Schwefel, H.-P. *Evolution strategies. A comprehensive introduction.* Natural Computing, 3-52, Kluwer Academic Publishers. Netherlands, 2002

[2] Castelino, K., D'Souza, R., and Wright, P. K. *Tool-path Optimization for Minimizing Airtime during Machining.* Journal of Computing and Information Science in Engineering, Volume 4, Issue 3, pp. 235-241, 2004

[3] Du, S. *Simulation and Tool Path Optimization for the Hexapod Milling Machine.* PhD, University of Dortmund. Vulkan Verlag, Essen, 2004

[4] Foley, J.D., Van Damm, A., Feiner, S. K., and Hughes, J. F. *Computer graphics, Principles and Practice.* Addision-Weseley Professional, 1995

[5] Gonzalez, R. C. and Woods, R. E. *Digital Image Processing 2. edition.* Prentice-Hall, New Jersey, 2002

[6] Hudson, T. C., Lin, M. C., Cohen, J., Gottschalk, S., and Manocha, D. *V-COLLIDE: Accelerated Collision Detection for VRML*. Second Symposium on the Virtual Reality Modeling Language, New York City, NY, February 1997, 119-125, 1997

[7] Huhse, J. and Zell, A. *Evolution Strategy with Neighborhood Attraction – A Robust Evolution Strategy.* Proceedings of the Genetic and Evolutionary Computation Conference, 2001, Edited by Spector, L. Goodmann, E. et al.

[8] Lin, M. C. and Gottschalk, S. *Collision detection between geometric models: a survey*. IMA Conference on Mathematics of Surfaces, Winchester, UK, 33-52, 1998

[9] Mortenson, M. E. *Geometric modelling.* John Wiley & Sons, Inc., 1985

[10] My, C. A., Bohez, E. L .J., Makhanov, S. S., Munlin, M., Phien, H. N., and Tabucanon, M. T. *On 5-Axis Freeform Surface Machining Optimization: Vector Field Clustering Approach.* International Journal of CAD/ CAM, Vol. 5, No.1, pp. 1598-1800, 2005

[11] Niemöller, A. *The wavelet transform as a tool in chemometrics: applications in NIR-spectrometry.* PhD, University of Duisburg. Department of Chemistry, 1999

[12] Nievergelt, Y. *Wavelets Made Easy*. Birkhauser Verlag AG, Juni 1999

[13] Schwefel, H.-P. *Evolution and Optimum Seeking.* Sixth-Generation Computer Technology, John Wiley and Sons, New York, 1995

[14] Stautner, M. and Zabel, A. *Optimizing the Multi-Axis Milling Process via Evolutionary Algorithms.* Proceedings of the 8th CIRP International Workshop on Modeling of Machining Operations, Chemnitz, Deutschland, 2005

[15] Surmann, T., Kalveram, M., and Weinert, K. *Simulation of Cutting Tool Vibrations for the Milling of Free Formed Surfaces.* Proceedings of the 8th CIRP International Workshop on Modeling of Machining Operations, Chemnitz, Deutschland, 2005

[16] Weinert, K., Mehnen, J., and Stautner, M. *The Application of Multiobjective Evolutionary Algorithms to the Generation of Optimized Tool Paths for Multi-Axis Die and Mould Making.* In: Intelligent Computation in Manufacturing Engineering, 4th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, CIRP ICME '04, Sorrento, Naples, pp. 406-412, 2004

[17] Weinert, K. and Stautner, M. *An Efficient Discrete Simulation for Five-Axis Milling of Sculptured Surfaces.* Production Engineering – Research and Development, Annals of the German Academic Society for Production Engineering, IX (2002)1, pp. 47-51, 2002

[18] Weinert, K. and Stautner, M. *Generating Multiaxis Tool Paths for Die and Mold Making with Evolutionary Algorithms.* In: Conference Proceedings Pt. II; Genetic and Evolutionary Computation – GECCO 2004, Seattle, WA, USA, pp. 1287-1298, 2004

[19] Weinert, K and Surmann, T. *Geometric Simulation of the Milling Process for Free Formed Surfaces.* In: Simulation Aided Offline Process Design and Optimization in Manufacturing Sculptured Surfaces, Witten Bommerholz, pp. 21-30, 2003

[20] Weinert, K. and Zabel, A. *Modelling, Simulation and Visualization of Simultaneous Five-Axis Milling with a Hexapod Machine Tool.* In: Simulation in Industry, 13th European Simulation Symposium, ESS01 Conference Proceedings, 18.-20. October 2001, Marseille, France, pp. 344-348. [SCS] Publication, Erlangen

[21] Vidkovic, B. and Mueller, P. *Wavelets for kids – A Tutorial Introduction*. Tutorial, Duke University, 1991

[22] Yoon, J.-H. *Tool tip gouging avoidance and optimal tool positioning for 5-axis sculptured surface machining.* International Journal of Production Research, 41(10):2125-2142, 2003

[23] Youn, U.W., Jun, Y., and Park, S. *Interference-free tool path generation in five-axis machining of a marine propeller.* The International Journal of Production Research, 41(18):4383-4402, 2003