Real-Time Interpolators for Multi-Axis CNC Machine Tools

Rong-Shine Lin and Yoram Koren

Department of MEAM
The University of Michigan
Ann Arbor, MI 48109-2125

Abstract

This paper presents a new real-time interpolation architecture for CNC machine tools. The real-time approach calculates new commands for the control loops during a short period of one sampling time. This makes the real-time interpolator capable of generating accurate references for the axes and eliminates unneeded in-segment accelerations and decelerations. Consequently, it produces smoother part surfaces and requires less machining time compared with conventional off-line approaches. With the new type of real-time interpolator, cutting a curve or surface requires only one NC g-code instruction. Therefore, the size of the cutter location (CL) file is significantly reduced.

1. Introduction

Integrated Computer-Aided Design/ Computer-Aided Manufacturing (CAD/CAM) systems interfaced with Computer Numerical Controlled (CNC) machine tools have been widely used in industry for manufacturing automation [Tseng, 1990; Lee and Chang, 1991; Hanada 1993]. These systems enable the production of a part surface from geometric models. In this process, the geometric data have to be transferred from the CAD/CAM to the CNC.

In the conventional method, the surface is decomposed by the CAD/CAM system into a sequence of straight lines that approximate the desired part surface with a given tolerance. These straight lines are converted into the cutter location (CL) file. The CL file actually describes the cutter path in a sequence of tool positions. Using the post processor, the CL file is finally converted into g-code straight-line instructions that are downloaded into the CNC system [Kawabe et al., 1981; Koren, 1983; Kato et al., 1984; Loney and Ozsoy, 1987; Choi et al., 1988; Hwang, 1992]. The reason for using this approach is that most conventional CNC machine tools provide only linear and circular interpolators and the linear fitting is the simplest approximation. The drawbacks of this off-line approach are:

a. it requires acceleration and deceleration steps at each line of the CL file, thereby producing less smooth curves and substantially increasing the cutting time;

b. tool orientations in 5-axis machining are interpolated inaccurately that subsequently causes position errors and unsmooth surfaces [Lin and Koren (a); 1994];

c. the size of the CL file will be very large for complicated parts that causes memory shortage problems and data transmission errors.

With increasing demands for improved machining precision, higher productivity, and greater versatility, methods for improving the performance of machine tools are continuously being sought. One of these methods is the development of real-time interpolators for multi-axis CNC machine tools [Shpitalni et al; 1994]. The real-time interpolator, which is contained in the CNC computer, calculates new commands for the control loops during a short time period (e.g., 1 to 10 msec). The successive commands are calculated during the execution time of the current commands. This approach produces smoother surfaces and requires substantially less machining time compared with the conventional off-line approach.

CNC system

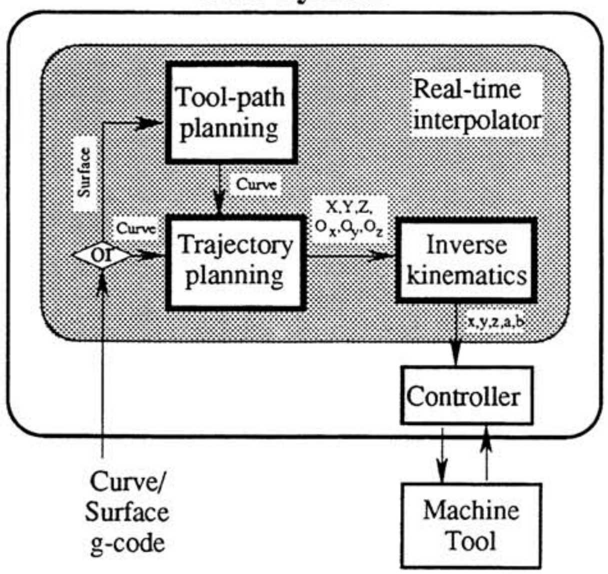

Figure 1: The scheme of the proposed real-time interpolator.

2. The Proposed Interpolator

The proposed real-time interpolator is based on the reference word interpolator method [Koren, 1983]. Figure 1 shows the scheme of the proposed real-time surface interpolator. This real-time interpolator consists of (i) tool-path planning, (ii) trajectory planning, and (iii) inverse kinematics. The input to the interpolator is a special g-code (i.e., NC instructions) of either a curve or a surface.

For surface machining, the interpolator begins with tool path planning, which generates curves as tool-paths, and subsequently executes the trajectory planning portion for the interpolation of each curve. When curve machining commands are specified, the interpolator starts with trajectory planning, which generates the cutter position (X,Y,Z) and orientation (O_X,O_Y,O_Z) of each interpolation step, for a constant feedrate. For 5-axis machines, the interpolator needs an additional portion, the inverse kinematics transformation, to generate the five axial reference positions (x,y,z,a,b), which, in turn, are transmitted to controllers for controlling the machine tool. For 3-axis machines, the inverse kinematics transformation is an identity matrix. The calculations for the tool positions\ orientations and the solution for the inverse kinematics have been presented in [Lin and Koren (b); 1994].

The proposed real-time surface interpolator generates the tool position and orientation (the latter in five-axis machining) in each sampling period, and sends the reference commands to the machine controller during cutting. Therefore, the proposed interpolator has the following advantages:

a. it produces smooth surfaces and reduces machining time by eliminating intermediate acceleration and deceleration steps;

b. it generates precise reference positions for 5-axis machining by accurately interpolating tool positions and orientations [Lin and Koren (a); 1994];

c. it eliminates the computer memory shortage problems by significantly reducing the size of the CL files.

3. The Structure of CNC Systems

The CNC system is a real-time computer controlled system. In each sampling time, the computer has to execute interpolation, servo-controller, and various compensation algorithms. Typically, the sampling time is between 1 and 10 msec, depending on the complexity of the system, the control computer, and the system application.

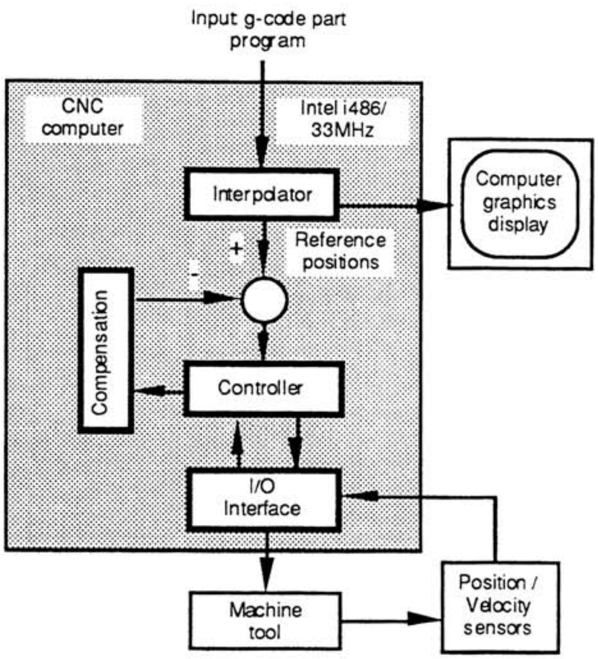

Figure 2: The structure of CNC system.

The 5-axis CNC system, on which the proposed real-time interpolators were implemented, has a sampling period of 10 msec. This system consists of a 5-axis milling machine controlled by an Intel i486/33MHz personal computer, I/O interfacing boards, and sensors, such as rotary encoders, linear encoders, tachometers, and motor current measurements. The structure of this CNC system is shown in Fig. 2. The input for the system is a g-code part program. According to the specific g-codes, the interpolator generates the reference positions for the controller. The controller compares the references with the actual position and velocity feedbacks, which are read from sensors, to generate the control signals. By means of the I/O interfacing boards, these signals are transmitted to the milling machine, thus driving the tool to desired positions.

The software program for this CNC system is written in C++ programming language. This program consists of a user input interface, a module for reading/decoding g-code part programs, interpolators, controllers, and hardware interfaces. Users can easily select the input parameters, such as the part programs, controllers, and sensor types, from the graphic pull-down window menu. First, the machine is initialized according to the chosen system parameters. Then, the part-program is read. The part program specifies what kind of interpolator is required. Each interpolator is programmed as a software subroutine. In accordance with the interpolation algorithm, the reference positions are generated, and sent to the servo-controllers. The program reads the actual table positions and velocities from sensors, compares them with the references, and transmits the control signals to the I/O interfaces. The system has a special feature of plotting in real-time the produced part by using the position feedbacks and the known part geometry.

In this system, the time needed for interfacing between the computer and the machine, such as reading position/velocity feedbacks and transmitting the control signals, and the control algorithms is 1 msec. The computation time for the proposed real-time 5-axis surface interpolator is 1 msec. The rest of the sampling time, 8 msec, deals with computer graphics and animation for interpolated tool positions display.

4. Relationship between Interpolation Accuracy and Sampling Time

The interpolation accuracy depends on the machining feedrate, the system sampling time, and the part radius of the curvature. Shown in Fig. 3 is an interpolated curve C that is approximated by a straight-line path L. The instantaneous radius of curvature of this curve is ρ. The error between the actual curve and the approximated straight line is the contour error and denoted as ε. From the Pythagoras theorem, we obtain

$$\left(\frac{L}{2}\right)^2 + (\rho - \varepsilon)^2 = \rho^2 \tag{1}$$

By rearranging,

$$\varepsilon = \frac{L^2}{8\rho} \tag{2}$$

where we assume that ε^2 is very small. For constant feedrate machining, the cutter moves a distance, VT, during each sampling time where V is the machining feedrate and T is the sampling time. For one sampling period, VT equals L. By substituting VT into Eq. (2) we obtain

$$\varepsilon_{max} = \frac{(VT)^2}{8\rho_{min}} \tag{3}$$

The system sampling period T is constant. If the feedrate V is constant, the maximum contour error is located at the area that has the smallest radius of curvature.

The parameter (VT) is the position resolution of the interpolation. We must keep it as small as possible to enhance interpolation accuracy. The feedrate is specified by the part programmer depending on the cutting conditions. The radius of the curvature represents the geometric properties of the desired curve. If these two conditions are given, the contour error is proportional to the square of the sampling time. In other words, the shorter the sampling period, the more precise the interpolation result will be. If we assume that the time spent on interfacing and the control algorithms is fixed (which is 1 msec in our CNC system), a faster interpolation algorithm will result in a more accurate curve interpolation.

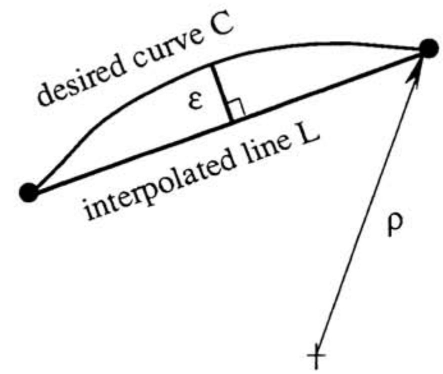

Figure 3: Linear approximation of a desired curve

For example, comparing an interpolation algorithm A that needs 1/2 msec computation time with an algorithm B, which requires 1 msec. With the same interface and controller (which need 1 msec) the systems A and B require sampling times of 1.5 and 2 msec, respectively. It is assumed that these two CNC systems do not require additional computation work, such as computer graphics display. With the same machining feedrate and the same curve, system A can interpolate 44% more accurately than system B.

The same advantage can also obtain if one system uses a computer which is 50% computationally faster than the other. With the same interpolation algorithm, the system with the faster computer can interpolate 44% more accurately than the slower one.

The above contour error analysis, based on Eq. (3), is valid when the interpolated points are located on the desired curve. For interpolation algorithms, which do not guarantee that the interpolated points are located on the desired curve (e.g., stairs approximation method by Danielson [1970]), the maximum contour error does not occur at the middle point of the approximate straight line. Therefore, Eq. (3) is not valid, and all contour errors along the curve have to be examined. After the contour errors are examined to satisfy the allowable tolerance, the feedrate errors also must be checked. As a ruled of thumb, feedrate errors must be less than 2% of the specified feedrate, in order to produce smooth curves.

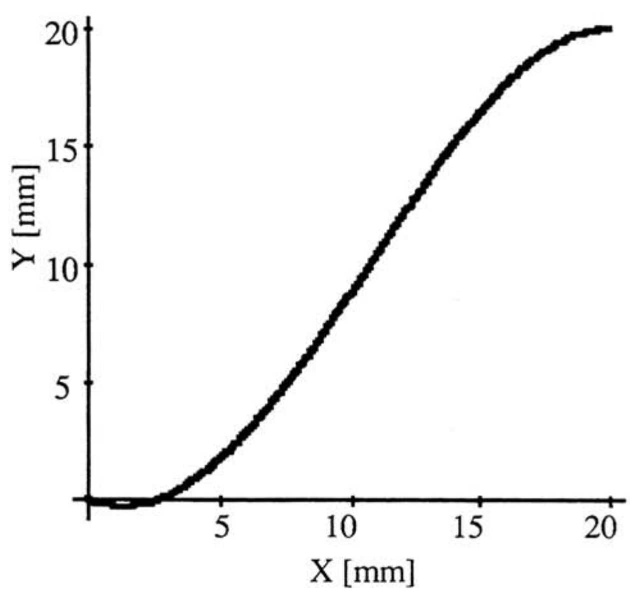

Figure 4: The desired curve of Eq. (4).

5. Discussions

In this section, a comparison of the real machining performance of the proposed interpolator and the off-line interpolator is presented. The example shown in Fig. 4 is a 2-D curve that was programmed and produced on our CNC machine by two methods: our real-time method and the off-line interpolator. The curve is described by

$$\begin{cases} x(u) = 20u \\ y(u) = -50u^3 + 80u^2 - 10u \end{cases} \quad 0 \le u \le 1. \quad (4)$$

This curve is programmed to be machined at a feedrate of 10 mm/sec. The result of this experiment is shown in Fig. 5. The real machining feedrate V is obtained by calculating, in real-time, the resultant velocity $V = \sqrt{V_x^2 + V_y^2}$ where V_x and V_y are the X- and Y-table velocities, respectively. These two table velocities are read from tachometers, which are installed on the X and Y motors.

For comparison, we programmed for the same curve also on off-line interpolator approximated by 92 straight lines programmed using g01 code. The machining velocity profile for this experiment is shown in Fig. 6. From Fig. 6, we see that the machine does not reach the programmed feedrate during most of the machining time. This is because the machine has to accelerate and decelerate at each of the 92 lines. The acceleration and deceleration steps also increase the machining time. By measuring the machining time, the proposed real-time approach requires 3.0 sec to finish the curve; however, the off-line approach needs 10.4 sec (10.1 sec cutting time and 0.3 sec for reading and decoding 92 g-code commands) which is 3.5 times more than that of the real-time interpolator. In addition, the size of the CL file is significantly reduced by our method.

we would like to emphasize that the system response depends not only on the interpolator type, but also on the servo-controller and the selection of its gains. For example by increasing the P-gain, I-gain, and D-gain of the PID controller from (2, 2, 15) to (6, 50, 25) we obtained the velocity response given in Fig. 7 and Fig. 8 for the two types of interpolators. Note the large overshoots that were obtained. This demonstrates that the interpolator should be designed concurrent with the servo-control loops.

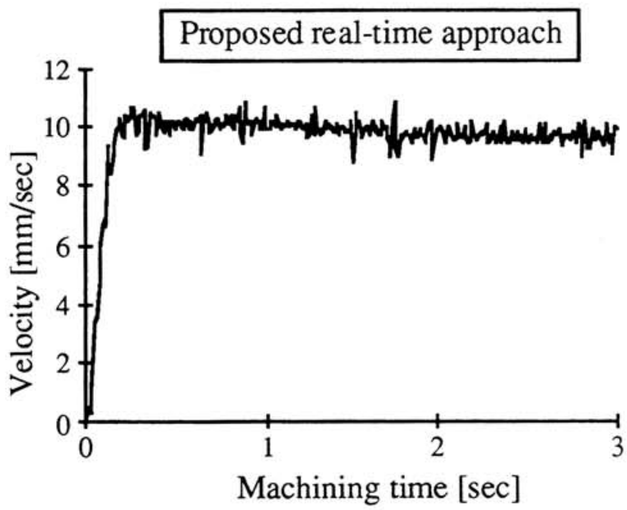

Figure 5: The measured feedrate for the proposed real-time approach.

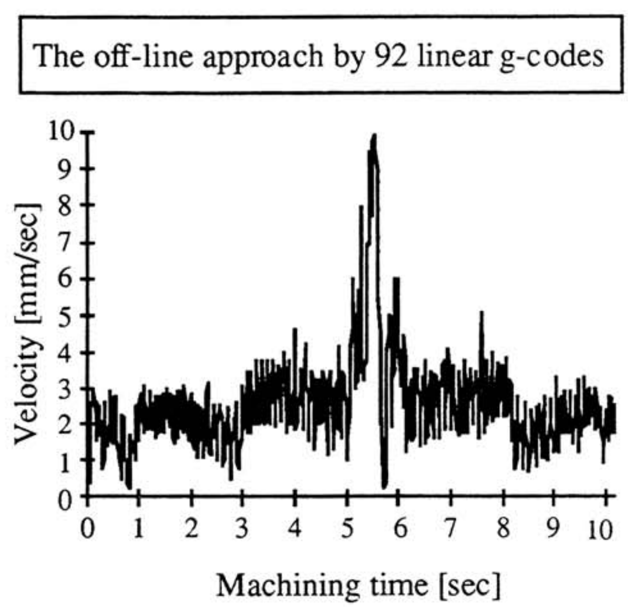

Figure 6: The measured feedrate for the off-line approach.

Finally, it is worthwhile to mention that the proposed real-time surface interpolator assumes that the part surface is gouging free. A gouging problem is occurred when the cutter radius is larger than the radius of a concave surface curvature (called local gouging) or when the cutter collides with the other part of surface (called global gouging). Several algorithms were proposed to detect and avoid gouging problems [Hwang, 1992; Jensen and Anderson, 1992; Renker, 1993]. However these algorithms fit only the off-line approach, because they require a very large computation load. We propose, therefore, that to obtain a gouging free surface the part surface will be examined in off-line before machining it with our real-time interpolator.

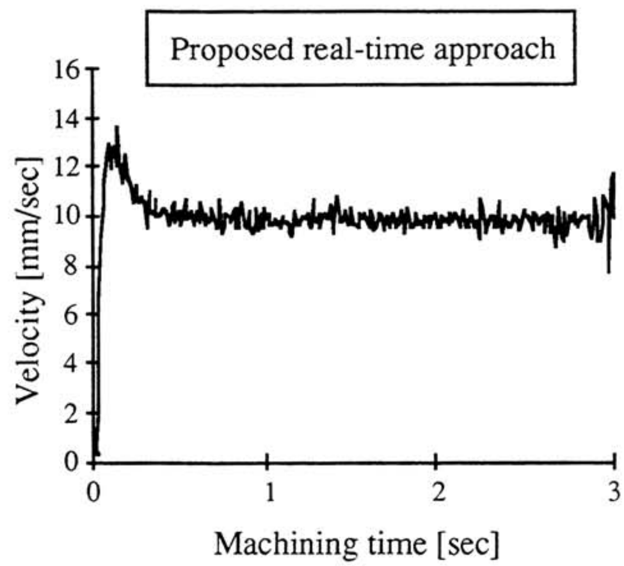

Figure 7: The measured feedrate for the real-time approach.

7. Conclusions

The proposed real-time interpolator has been demonstrated and implemented on a 5-axis CNC milling machine. The results demonstrated that the real-time approach has several advantages over the conventional off-line approach. With the proposed real-time approach, the CNC machine can eliminate the intermediate acceleration and deceleration steps. This provides a constant feedrate machining and produces a smooth surface. This also reduces machining time. Using the proposed real-time interpolator, machining a curve or surface requires only one g-code instruction; therefore, the size of a CL file is significantly reduced.

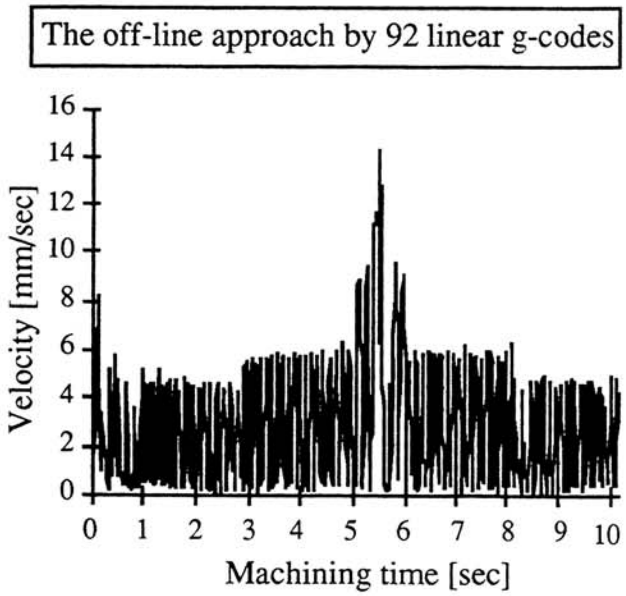

Figure 8: The measured feedrate for the off-line approach.

References

Choi, B., Lee, C., Hwang, J., and Jun, C., 1988, " Compound Surface Modeling and Machining", Computer-Aided Design, Vol 20, no 3, pp 127-136.

Danielsson, P., 1970, " Incremental Curve Generation", IEEE Trans. on Computers Vol c-19, no 9, Sep., pp 783-793.

Hanada, T., Bandyopadhyay, B. P., and Hoshi, T., 1993, " Development of A CAD/CAM System for Rapid Prototyping," ASME PED-Vol. 64, Manufacturing Science and Engineering, pp 101-108.

Hwang, J., 1992, "Interference-free tool-path generation in the NC machining of parametric compound surfaces", *CAD* Vol. 24, no 12, Dec., pp 667-677.

Jensen, C. and Anderson, D., 1992, "Accurate Tool Placement and Orientation for Finish Surface Machining", *ASME winter annals meeting*, pp. 127-145.

Kawabe, S., Kimura, F., and Sata, T., 1981, "Generation of NC Commands for Sculptureed Surface Machining form 3-Coordinate Measuring Data", CIRP Ann., Vol 30, no 1, pp 369-372.

Koren, Y., 1983, "Computer Control of Manufacturing Systems", McGraw-Hill Book Company, New York.

Kota, K., Oshima, M., and Maruyama, H., 1984, "The DDI-CNC for Machining Sculptured Surfaces", IEEE Conference on IECON, New Jersey, pp 551-555.

Lee, Y. and Chang, T., 1991, " CASCAM-An Automated System for Sculptured Surface Cavity Machining", *Computers in Industry* Vol. 16, pp. 321-342.

Lin, R. S. and Koren, Y. (a), 1994, " Real-Time Five-Axis Interpolator for Machining Ruled Surfaces," ASME winter annual meeting, Chicago, November, DSC-Vol. 55-2, pp 951-960.

Lin, R. S. and Koren, Y.(b), 1994, " Real-Time Interpolator for 5-Axis Surface Machining," The 3rd International Conference on Automation Technology, Taipei, July, pp. 107-113.

Loney, G. and Ozsoy, T., 1987, " NC Machining of Free Form Surfaces", Computer-Aided Design, Vol 19, no 2, pp 85-90.

Renker, H., 1993, "Collision-Free Five-Axis Milling of Twisted Ruled Surfaces", Annals of the *CIRP* Vol. 42/1/1993.

Shpitalni, M., Koren, Y., and Lo, C., "Real-Time Curve Interpolators," 1994, Computer-Aided Design, Vol. 26, Nov. pp 832-838.

Tseng, A., Kolluri, S., and Radhakrishnan, P., 1990, "A CNC Machining System for Education", Journal of Manufacturing System, Vol 8, no 3, pp 207-214.