Curve fitting with arc splines for NC toolpath generation

Millan K Yeung and Desmond J Walton

Many CAD/CAM systems use linear interpolation in the generation of numerical control toolpaths to approximate noncircular curves or profiles that are represented by a large number of points. This method induces processing problems when the dataset is large and overdetermined. The paper introduces a method of generating an NC toolpath with an arc spline and reducing the overdetermined data simultaneously.

Keywords: numerical control, toolpaths, splines

CAD/CAM systems are widely used in today's manufacturing industries. Designers use CAD systems to design parts for visual and theoretical analysis. On the basis of the CAD model, the numerical control (NC) programmer uses the CAM system to generate an NC toolpath for a computerized numerical control (CNC) machine so that it can produce the part. Besides using a CAD model, some CAM systems* can process data acquired from other advanced data acquisition systems, such as laser scanners and computer tomography (CT) scanners, directly. The sets of data acquired from these systems are usually large and overdetermined. Overdetermination occurs when the base object can be represented by fewer data points than the number of data points acquired. Most of these CAM systems connect the data points with linear segments to generate an NC toolpath. This method creates many short linear move commands with sudden changes in direction. Toolpaths created in this way induce problems for the CNC machine during the machining process, and this can affect the finished quality of the part. To reduce these problems, we introduce a method that uses arc segments to generate an NC toolpath.

*For example, the Delta System II CAD/CAM system, developed by the Clynh Prosthetic & Orthotic Laboratory, Calgary, Alberta, Canada.

Institute for Advanced Manufacturing Technology, National Research Council Canada, Montreal Road, Ottawa, Ontario K1A 0R6, Canada
*Department of Computer Science, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada
Paper received: 21 May 1993. Revised: 4 October 1993

MACHINING PROBLEMS INDUCED BY LINEAR MOVE COMMANDS

A CNC machine is a piece of manufacturing equipment that performs machining automatically to produce parts. It is controlled by a computer which reads in a set of move commands and other control commands to direct the operation of the machine. This set of commands is called an NC program. It is generated by the NC programmer using a CAM system or other toolpath generation methods such as manual NC programming. In general, CNC machines usually have two movement modes, the linear interpolation mode, and the circular interpolation mode. With the linear interpolation mode, the control moves the cutter from point to point on a straight line path. With the circular interpolation mode, the control moves the cutter from point to point along a circular arc path.

When using the linear interpolation method to generate a toolpath on a large and overdetermined dataset, many short linear move commands are generated. These excessive move commands inflate the size of the NC program, which may sometimes exceed the storage capacity of the CNC controller so that the whole NC program cannot be stored in the controller at one time. As the distance between points in these overdetermined datasets is usually very short, the distance of the linear motion for each NC command is also shortened. This causes the data buffer in the CNC controller to underflow. This buffer underflow problem occurs when the machine completes the physical motion of the current command and is ready to execute the next one, but the next command has not yet been completely transmitted. The current cutting movement stops, and the tool waits for the next command before making the next move. In addition, the linear moves in the toolpath are tangentially discontinuous from one move to the next. This tangential discontinuity causes a sudden change of direction at each point along the toolpath. The stop-and-go motion and the change of cutting direction cause oscillation along the toolpath and affect the quality of the finished surface. Intense oscillation may actually damage the machine tool.

One way to reduce these problems is to assign a slower feedrate to the NC program so that the machine operation slows down, and allow the data transmission to catch up with the cutting motion. However, the change
of cutting direction problem remains, and the finish quality does not improve. A better method is required.

**ARC-SPLINE SOLUTION**

In order to reduce the excessive number of data points with minimal alteration of the original profile, these points must be removed heuristically and logically. "Tolerance dependent data-point elimination" can be used to reduce the excessive amount of data, but this uses linear segments to define the toolpath, and thus the problem of the change of cutting direction remains. To avoid the change of direction between moves in the toolpath, moves must be smooth. With these constraints, this paper proposes a method that uses an arc spline instead of linear line segments to generate an NC toolpath. An arc spline is a $G^1$ curve made of circular arcs and straight line segments. A $G^1$ curve is continuous with a continuous unit tangent. A toolpath generated with an arc spline consists of both linear and circular interpolation commands. The problems induced by a sudden change of cutting direction are eliminated because of the continuity of the unit tangent. The importance of the circular-arc cutting path has been addressed by the manufacturing industries. Furthermore, when applying a tolerance constraint to the arc spline, the excessive points within the tolerance are excluded from the NC toolpath.

There are many methods of defining a circular arc curve with $G^0$ and $G^1$ continuity on discrete data, but most of them require the user to specify how many arcs are to be used beforehand. The arc spline used in this paper does not require this information. An analysis was carried out by Meek and Walton. The following discussion is based on their results. An arc-spline segment is composed of a pair of circular arcs, called a biarc, and two tangents at the two endpoints of the segment. To define a biarc, some transformations of the data are required. These transformations must be reversed after the biarc has been defined.

**DEFINITION OF BIARC BETWEEN TWO ARBITRARY POINTS**

If two distinct points $A$ and $B$ and two unit tangents $T$ and $U$ at these points are given, the biarc is a $G^1$ curve segment made up of two circular arcs connecting $A$ and $B$, and its ends match $T$ and $U$. To maintain generality, transform $A$ to the origin, and rotate $T$ so that it is pointing along the positive $x$ axis. Both $B$ and $U$ are also transformed and rotated with respect to the transformation and rotation of $A$ and $T$. Let the counterclockwise angle between $T$ and $U$ be $W$, where $W$ is in $(-\pi, \pi)$. If $W < 0$, reflect $B$ across the $x$ axis so that one can assume that $W > 0$. Further, if $W = 0$ and $B$ is below the $x$ axis, reflect $B$ across the $x$ axis so that $B$ is on or above the $x$ axis. Let $x$ be the angle which the vector $B - A$ makes with the $x$ axis. Let the radii of the two circular arcs be $r_a$ and $r_b$, and let the centres of the two circles be $C_a$ and $C_b$.

There are two main cases and several subcases that must be considered to determine the type of biarc used to join the points.

**Case 1 ($W = 0$)**

If $B$ is on the positive $x$ axis, $x = 0$, a biarc is not possible, but the straight line segment joining $A$ to $B$ matches the tangents $T$ and $U$. If $B$ is above the $x$ axis, $0 < x < \pi$, the S-shaped type 0 biarc shown in Figure 1 joins $A$ to $B$ and matches the tangents $T$ and $U$, where $r_a = r_b = \|B - A\|/4 \sin x$.

**Case 2 ($W > 0$)**

The type of biarc to be used depends on the position of $B$. If $-\pi + 0.75W < x < -0.125W$, the type 1 biarc in Figure 2 is used. If $0.125W < x < 0.875W$, the C-shaped type 2 biarc in Figure 3 is used. If $0.875W < x + 0.25W$, the S-shaped type 3 biarc in Figure 4 is appropriate.
The factors 0.75, 0.125 and 0.825 do not have an obvious geometric interpretation. They were arrived at empirically in an attempt to balance the circular arcs so as to decrease the probability of the resulting arc spline having one large arc and a lot of smaller arcs.

The radii \( r_A \) and \( r_B \) for these three types of biarc can be determined from Reference 9 following some algebraic manipulation.

These cases and subcases cover most of the situations encountered in defining a biarc between two points. However, there are situations in which a biarc as described above cannot be found, but they occur rarely in toolpath generation for CNC machines. For example, if \( W = 0 \) and \( B \) is on the negative \( x \) axis, there is no biarc that joints \( A \) to \( B \) and matches \( T \) and \( U \). To deal with such a situation, simply join the points with a linear segment, and reduce the feedrate of the machine so that the effect of tangential discontinuity is minimized. Restore the feedrate after the segment has been machined.

### REDUCING EXCESSIVE NUMBER OF DATA POINTS

In order to reduce an excessive number of points that are within a given tolerance, the distance between a point and the biarc is required. The distance \( D \) from point \( P \) to a biarc joining points \( A \) and \( B \) can be defined as follows:

\[
D = \begin{cases} 
\| P - C_A \| - r_A & \text{if } P \text{ is in wedge } A \\
\| P - C_B \| - r_B & \text{if } P \text{ is in wedge } B \\
\min\{\| P - A \|, \| P - B \|\} & \text{otherwise}
\end{cases}
\]

where wedge \( A \) has its apex at \( C_A \) and wedge \( B \) has its apex at \( C_B \) as shown in Figure 1, Figure 2, Figure 3 or Figure 4. Wedge \( A \) has arms at angles \(-0.5\pi \) and \(2\alpha - 0.5\pi \), and wedge \( B \) has arms at angles \(2\alpha + 0.5\pi \) and \(0.5\pi \), for the type 0 biarc as shown in Figure 1. Wedge \( A \) has arms at angles \(0.5\pi \) and \(2\alpha + 0.5(\pi - W)\), and wedge \( B \) has arms at angles \(2\alpha - 0.5(\pi + W)\) and \(W - 0.5\pi \) for the type 1 biarc as shown in Figure 2. Wedge \( A \) has arms at angles \(-0.5\pi \) and \(\alpha - 0.5\pi \), and wedge \( B \) has arms at angles \(\alpha - 0.5\pi \) and \(W - 0.5\pi \), for the type 2 biarc as shown in Figure 3. Wedge \( A \) has arms at angles \(-0.5\pi \) and \(2\alpha - 0.5(\pi - W)\), and wedge \( B \) has arms at angles \(2\alpha + 0.5(\pi - W)\) and \(W + 0.5\pi \), for the type 4 biarc as shown in Figure 4.

To determine whether a point \( P \) is in the wedge as shown in Figure 5, set the arm \( a \) connecting the apex \( C_A \) and point \( A \) to be the \( x \) axis. Measure the angle counterclockwise from \( a \) to the other arm \( b \) to obtain the spanning angle \( \theta \) of the wedge. Now connect \( P \) to \( C_A \) and measure the angle \( \alpha \) from \( a \) to this line. If \( \alpha \) is less than or equal to \( \theta \), \( P \) is in the wedge; otherwise, \( P \) is outside the wedge.

### DEFINE BIARC TOOLPATH WITH LONGEST ARC SCHEME

Start at the first point \( P_s \) of the dataset. Find the biarc \( A~B \) connecting \( P_s \) to a point \( P_i \), where \( i > s \), is as large as possible, such that all the points between \( P_i \) and \( P_s \), i.e. \( \{P_{i+1}, \ldots, P_{i-1}\} \), are within a given distance (tolerance) from the biarc. Set \( P_i \) to be the new \( P_s \). Repeat the process until the entire dataset is exhausted.

The longest arc approach allows sequential processing of the input data, and is thus space efficient.

### EXAMPLES

#### Example 1

A sample part of a rotary cam was used to test the biarc toolpath. The outline profile of the cam is represented by 98 data points. In this test, three different definitions are used to define \( \theta \) for the three types of biarc in Case 2.

- For a type 1 biarc, \( \theta = 2\alpha + 0.5W \)
- For a type 2 biarc, \( \theta = 2\alpha - 0.5W \)
- For a type 3 biarc, \( \theta = -2\alpha + 0.5W \)

Since the profile is a closed curve, the following criterion is used to define the tangent \( V_i \) at each point \( P_i \):

\[
V_i = \frac{P_{i+1} - P_{i+1}}{\|P_{i+1} - P_i\|} + \frac{P_{i+1} - P_{i}}{\|P_{i+1} - P_i\|} \quad i = i \mod n + 1
\]
The reason for this choice of $V_i$ is the following. For

$$ \|P_i - P_{i-1}\| = \|P_{i+1} - P_i\| = \delta $$

$V_i$ tends to a vector which is tangential to a curve passing through $P_{i-1}$, $P_i$ and $P_{i+1}$ at $P_i$ in the limit as $\delta \to 0$. The same criterion is also applicable to the interior points of an open curve. The tangent vectors at the endpoints of an open curve can be approximated by fitting a quadratic Bézier curve with monotone curvature whose turning point occurs at the interior point adjacent to the endpoint. This yields

$$ V_i = P_2 - 0.5[V_2(P_2 - P_1)]V_2 $$

and

$$ V_n = P_{n-1} + 0.5[V_{n-1}(P_n - P_{n-1})]V_{n-1} $$

Two samples were machined. One used the linear toolpath defined using the original data without data reduction, and the other used the biarc toolpath with a data reduction of 0.001 in tolerance. The linear path is composed of 98 linear move commands, while the biarc path consists of only 30 circular-arc move commands (15 biarcs). Figure 6 shows the outline profile of the cam in both representations. The surface of the sample machined using a linear toolpath consists of 98 flat facets, and is very rough. The surface of the sample machined using a biarc path is very smooth. It is obvious that the biarc path is better than the conventional linear path.

**Examples 2 and 3**

Two more examples are shown in Figure 7. The profiles of an outlined letter C and a letter S were captured by digitization. The arc-spline process reduced the number of segments in both cases. A 0.01 tolerance was used for the reduction. 104 linear segments in the letter C were reduced to 70 circular segments, or 35 biarcs, and 128 linear segments in the letter S were reduced to 76 circular segments, or 38 biarcs. The profiles approximated by the circular arc segments are clearly smoother than the ones approximated by the straight line segments. This further confirms the practicality of using the arc spline in curve fitting for CNC machining.

**CONCLUSIONS**

Conventional NC programming uses linear segments to approximate a smooth curve in the generation of NC toolpaths. This method works, but there is a discontinuity of tangents problem between segments. This problem may be amplified if the number of data points used to generate the toolpath is large and overdetermined. The toolpath generated by this method induces storage and oscillation problems for the CNC machine.

A biarc toolpath based on arc-spline theory reduces these problems, and improves the finish quality of the machined part. It uses continuous circular-arc segments to approximate the smooth curve. With a given tolerance, it can remove the excessive data points, and thus reduce the size of the NC program. The continuous segments and data reduction overcome the problems induced by the linear path, and therefore a biarc toolpath should be used instead of a linear toolpath whenever it is applicable.
The technique discussed in this paper could be considered as curve fitting for NC toolpath generation. As it is a curve fitting method, it is desirable to consider the issues of segmentation, noise, and the representation of the final curve.

Segmentation

The method can produce unevenly placed segments, e.g. one large arc and many small ones. The choice of the biarc segmentation factors as discussed for Case 2 partially alleviates this problem. It is good practice to display the NC toolpath before the machining process. This allows the detection of unevenly spaced segments. If unevenly spaced segments are present, the data can be segmented at a point which would partition an existing large arc.

Noise

Input data from two different kinds of source are considered: first, the measurement of an existing object for the sake of replicating it, and, second, the use of a computer-aided design system in order to manufacture a designed object.

Data from the first source are subject to sampling error. It is assumed that the measurements would be made within a tolerance, e.g. using a coordinate measuring machine. This tolerance, as well as the tolerance used in defining the biarc toolpath with the longest arc, is determined by the tolerance within which replication is desired.

Data from the second source would only be subject to rounding error. In this case, a CAD output precision should be selected which is consistent with the desired tolerance for the manufacture of the designed object.

Representation of final curve

The motivation for the work presented here is the efficient and effective use of the circular arc cutting feature of CNC machines. Since this is the ‘final’ toolpath, it is not expected that another representation of the curve would be required; the data usually arises from a previous representation of the curve. However, in the case of replication, it may be desirable to put the curve into a Bézier format or other format if it is necessary to replicate a modified model of the original object. Rather than convert an arc spline to a Bézier format, B-spline format, NURBS format, or some other format, it may be more efficient to first fit a curve of the desired format to the data. Once modified, a new dataset can be obtained to which an arc spline can be fitted for final toolpath generation. There are also methods which can be used to convert curves in NURBS format or Bézier format to an arc spline form\(^1,10,11\).

ACKNOWLEDGEMENTS

The authors acknowledge with thanks some very useful discussions with Professor D S Meek, and the comments of the anonymous referees of this paper, which have helped to improve the final version of the paper.

REFERENCES

2 Beard, T 'In search of the perfect curve' Modern Mach. Shop (Jul 1991) pp 58–68
3 'The "no problem" problem' Industr. Laser Rev. (May 1992) pp 13, 14
6 Hoschek, J 'Circular splines' Comput.-Aided Des. (submitted for publication)
8 Sabin, M A 'Use of piecewise forms for numerical representation of shape' Report 60/1977 Computer & Automation Institute, Hungarian Academy of Sciences, Hungary (1977)

Millan Yeung is a research officer and group leader of the rapid response manufacturing group at the Institute for Advanced Manufacturing Technology of the National Research Council of Canada. The focus of his research is on applying computer technologies to manufacturing; these include AI, CAD/CAM and CNC. He obtained an MSc from the University of Manitoba in 1992. He is enrolled in the PhD program at the Ottawa–Carleton Institute of Electrical Engineering, University of Ottawa and Carleton University, Canada.

Desmond Walton is a professor of computer science at the University of Manitoba, Canada. His research interests include the application of computer graphics techniques and numerical analysis to problems in CAD/CAM. He obtained a PhD from the University of Manitoba in 1978. Dr Walton spent a year on research leave at the National Research Council of Canada, where he became interested in the application of splines to problems in toolpath generation.