

Research and Development of Sampled-data Interpolation Algorithm Software in CNC System Based on the Visual C++

Linguo YANG¹, Guozhen ZHANG²

¹Anhui Vocational and Technical College ,Hefei, 230011, China

²Anhui Technical College of Mechanical and Electrical Engineering, Wuhu, 241002, China

Abstract: Nowadays, sampled-data interpolation algorithm has been the core technology in modern CNC system, so there are more researches concerning description of the principle of the arc-interpolation and presentation of the flowchart of the arc sampled-data interpolation algorithm. With the development of virtual simulation system software, the coordinate of arc-center is a key question inferred and confirmed by formulas. Therefore, the method of affine transformation is presented to transform the coordinates of the arc or the spiral curve. In addition, some major and minor arc, the clockwise and counterclockwise arc of quadrants, spiral curve along with these sampled-data interpolation algorithm software will be displayed in CNC virtual simulation system based on the visual C++.

1. Introduction

CNC machine-tool is the key equipment in modern manufacturing industry, and CNC path control (interpolation) is the core principle of CNC machine-tool, which reveals the nature of CNC machining. In traditional CNC system, the interpolation algorithm can be divided into two categories: pulse incremental interpolation algorithm and sampled-data interpolation algorithm, both of which can be used to implement the interpolation of spatial line, spiral curve or arc, and some of algorithms can be used to implement the interpolation of spline or parabola. The principles and the subsequent improvement of these algorithms have been mature^[1]. As we know, the interpolation algorithm is an implemented with hardware to meet the need of real-time^[2], however, with the development of the high-speed of computer software, the interpolation algorithm can be implemented by software to meet the real-time requirement. It is the hot topic of path control in CNC system, high-precision and high-speed of the interpolation algorithm and implemented are researched in many results^[3-5]. However, there are few researches about CNC virtual simulation system, especially software development with using visual C++. In this thesis, the author attempts to elaborate the introduction of the principle of sampled-data interpolation algorithm and the establishment of the CNC virtual simulation software platform.

In the following sections, section two depicts the principle of the sampled-data interpolation algorithm for arc-interpolation; section three analyzes the key technology of software development; section four demonstrates examples of software development based on Visual C++ and section five offers a conclusion.



2. Principle of Sampled-data Interpolation Algorithm for Arc-interpolation

In order to meet the machining precision, the string or secant is used, instead of which curve to realize feed, namely linear approximating arc. The principle of inscribed linear is to describe the string between the arc and two sampling points and approximate the corresponding one. The $A(X_{i-1}, Y_{i-1})$ and $B(X_i, Y_i)$ are supposed the two adjacent interpolation points of the arc, the linear AB values of the arc \widehat{AB} equal to ΔL , the formula is:

$$\Delta L = FT_s \tag{1}$$

Where “F” means Feed, “Ts” means interpolation cycle.

When the tool moves from A to B, the X axis coordinate increment values is a $|\Delta X_i|$, and the Y axis coordinate increment values is a $|\Delta Y_i|$. See Fig.1. Compare of $|X_{i-1}|$ and $|Y_{i-1}|$ values, these fomulas are infered as follows:

When the $|X_{i-1}| < |Y_{i-1}|$:

$$\begin{cases} \Delta X_i = \frac{\Delta L}{R} \left(Y_{i-1} + \frac{1}{2} \Delta Y_{i-1} \right) \\ \Delta Y_i = -Y_{i-1} \pm \sqrt{R^2 - (X_{i-1} + \Delta X_i)^2} \end{cases} \tag{2}$$

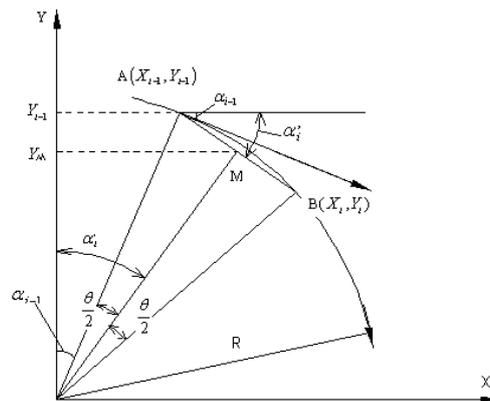


Fig.1 Sampled-data interpolation algorithm for arc-interpolation former.

When the $|X_{i-1}| \geq |Y_{i-1}|$:

$$\begin{cases} \Delta X_i = \frac{\Delta L}{R} \left(Y_{i-1} + \frac{1}{2} \Delta Y_{i-1} \right) \\ \Delta Y_i = -Y_{i-1} \pm \sqrt{R^2 - (X_{i-1} + \Delta X_i)^2} \end{cases} \tag{3}$$

The moving point coordinate is expressed as:

$$\begin{cases} X_i = X_{i-1} + \Delta X_i \\ Y_i = Y_{i-1} + \Delta Y_i \end{cases} \tag{4}$$

In the formula(2) and (3), the “±” Selection and arc where in area of the relevant quadrant, which divided into 4 quadrant areas by linear of $Y=X$ and $Y=-X$, in the XOY place, there are areaI, areaII, areaIII, areaIV. See Fig.2.

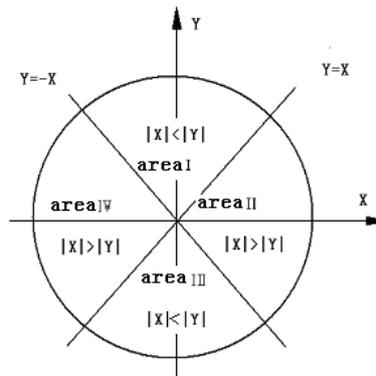


Fig.2 Divided area in the XOY place former.

The formula of areaI:

$$\Delta Y_i = -Y_{i-1} + \sqrt{R^2 - (X_{i-1} + \Delta X_i)^2} \tag{5}$$

The formula of areaII:

$$\Delta X_i = -X_{i-1} + \sqrt{R^2 - (Y_{i-1} + \Delta Y_i)^2} \tag{6}$$

The formula of areaIII:

$$\Delta Y_i = -Y_{i-1} - \sqrt{R^2 - (X_{i-1} + \Delta X_i)^2} \quad \Delta X_i = -X_{i-1} - \sqrt{R^2 - (Y_{i-1} + \Delta Y_i)^2} \tag{7}$$

The formula of areaIV:

$$E_i = (X_i - X_e)^2 + (Y_i - Y_e)^2 \leq (FTs)^2 \tag{8}$$

By contrast, the formulas from (5) to (7), and the formulas from (6) to (8), Only through interpolation formula (5) of the area I and formula (6) of the area II are sufficient, the area III and the area IV can be achieved by conversion symbol mark S_1 . If considered further counterclockwise arc-interpolation, it's necessary to introduce another symbol mark S_2 to convert, so that the arc interpolation flowchart is shown in Fig.3.

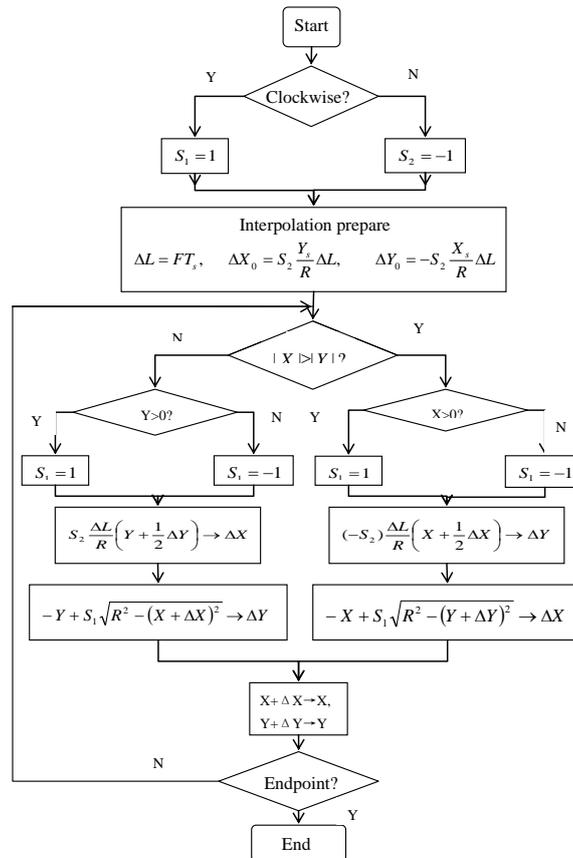


Fig.3 The arc-interpolation flowchart.

In the Fig.3, the conditions of the endpoint is:

$$\Delta L = 2\sqrt{2 Re_r - e_r^2} \leq \sqrt{8 Re_r} \tag{9}$$

Where the E_i is the distance between the current inter-polation point (X_i, Y_i) and its endpoint (X_e, Y_e) .

When the moving point arrives at the end of the profile curve at once, and then the corresponding signs will be set in system, and process the next section contour.

The formula of interpolation error as follows:

$$\Delta X_i = -X_{i-1} - \sqrt{R^2 - (Y_{i-1} + \Delta Y_i)^2} \tag{10}$$

Where the “ e_r ” is the maximum radial error-tolerance.

If the maximum allowable radial error is $e_r \leq 1\mu m$, and the interpolation cycle is $T_s = 8ms$ and the unit of feed speed is mm/min , these values are brought into the formula (10), and the feed speed F can be obtained as:

$$F \leq \sqrt{450000} R (mm/min) = 671R (mm/min) \tag{11}$$

3. Key Technology of Software Development

3.1 Circular Center Coordinate of Arc

In the 2-D coordinate system, the θ value is confirmed by arc start point $(x1, y1)$ and endpoint $(x2, y2)$, see Fig.4.

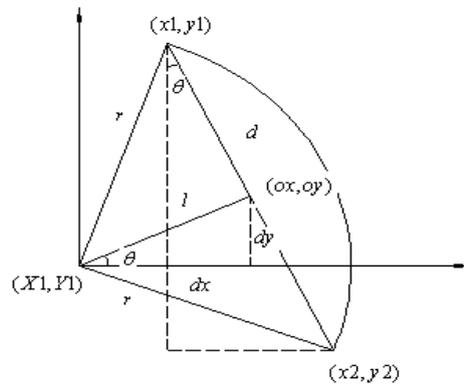


Fig.4 The determining arc center.

$$\begin{cases} \cos \theta = (y1 - y2) / d \\ \sin \theta = (x1 - x2) / d \end{cases} \tag{12}$$

$$\begin{cases} dx = l \cdot \cos \theta \\ dy = l \cdot \sin \theta \end{cases} \tag{13}$$

$$\begin{cases} ox = (x1 + x2) / 2 \\ oy = (y1 + y2) / 2 \end{cases} \tag{14}$$

$$\begin{cases} X1 = ox + dx \\ Y1 = oy + dy \end{cases} \tag{15}$$

However, the start point and the end point of the arc can't be presented accurately by the formula (15), there are 12 situations, which are shown in Fig.5. So, two variable cx and cy to denote in which side the center lies at the arc chord, the numbers “-1, 0, 1” are used for locating the positions.

Definition:

$$ox = (x1 + x2) / 2, oy = (y1 + y2) / 2.$$

Regardless of G02 of clockwise or G03 of anticlockwise interpolations, if you interchange the start point with end point of G02 clockwise interpolation coordinate, the G03 counterclockwise interpolation will be fastened. In the table1 shows the G02 clockwise interpolation coordinate.

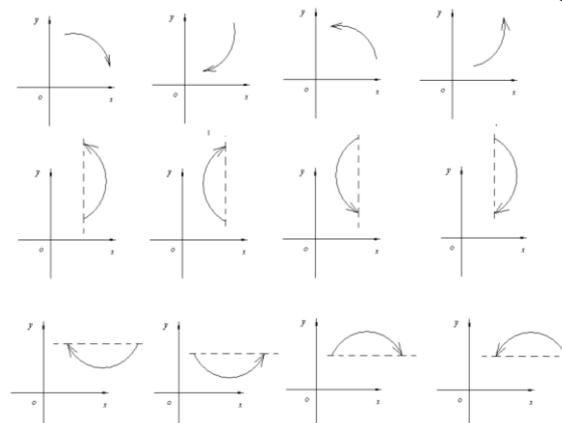


Fig.5 The 12 different situations of arc.

Table 1: The arc center of G02 clockwise interpolation coordinate

Comparis on of $x1,$ $x2$	Comparis on of $y1,y2$	cx	cy	X1	Y1
$x1>x2$	$y1>y2$	1	-1	$ox+dx$	$oy-dy$
$x1>x2$	$y1=y2$	0	-1	ox	$oy-dy$
$x1>x2$	$y1<y2$	-1	-1	$ox-dx$	$oy-dy$
$x1=x2$	$y1>y2$	1	-1	$ox+dx$	$oy-dy$
$x1=x2$	$y1=y2$	-1	1	$ox-dx$	$oy+dy$
$x1=x2$	$y1<y2$	-1	1	$ox-dx$	$oy+dy$
$x1<x2$	$y1>y2$	1	1	$ox+dx$	$oy+dy$
$x1<x2$	$y1=y2$	0	1	ox	$oy+dy$
$x1<x2$	$y1<y2$	-1	1	$ox-dx$	$oy+dy$

3.2 Coordinate Transformation for Arc

The counterclockwise arc AB has been illustrated by four quadrants “3, 4, 1 and 2” in the XOY coordinate system, but the circular center of the arc was not in the coordinate origin “ O ”. See Fig.6. The proposal of the coordinate transformation method solves this kind of the arc interpolation, the arc center is built in the new coordinate system $X'O'Y'$, every axis of which is parallel to the axis of XOY coordinate system, and the arc AB is divided into two parts that are respectively the arc AS and the arc SB . Taking the arc SB for example, the coordinates of the point S, B, C are defined as $(x_S, y_S), (x_B, y_B), (x_C, y_C)$ which in the XOY coordinate system, application of matrix formula (16), the coordinates of the point $S(x_S, y_S)$ and the point $B(x_B, y_B)$, which in the $X'O'Y'$ coordinate system can be obtained as:

$$\begin{pmatrix} x'_S & y'_S \\ x'_B & y'_B \end{pmatrix} = \begin{pmatrix} x_S & y_S \\ x_B & y_B \end{pmatrix} - \begin{pmatrix} x_C & y_C \\ x_C & y_C \end{pmatrix} \quad (16)$$

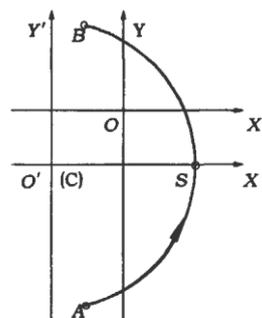


Fig.6 Coordinate transformation of arc in different quadrant.

The arc SB interpolation in the XOY coordinate system can be implemented by calling subprogram, the middle point coordinate values of the arc SB can be obtained as follows matrix:

$$\begin{pmatrix} x'_S & y'_S \\ x'_1 & y'_1 \\ x'_2 & y'_2 \\ x'_3 & y'_3 \\ \dots & \dots \\ x'_{n-1} & y'_{n-1} \\ x'_n & y'_n \\ x'_B & y'_B \end{pmatrix} \tag{17}$$

Application of the formula (17), the middle point coordinate values of the arc SB that in the XOY coordinate system can be obtained as:

$$\begin{pmatrix} x_S & y_S \\ x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \\ x_{n-1} & y_{n-1} \\ x_n & y_n \\ x_B & y_B \end{pmatrix} = \begin{pmatrix} x'_S & y'_S \\ x'_1 & y'_1 \\ x'_2 & y'_2 \\ x'_3 & y'_3 \\ \dots & \dots \\ x'_{n-1} & y'_{n-1} \\ x'_n & y'_n \\ x'_B & y'_B \end{pmatrix} + \begin{pmatrix} x_C & y_C \\ x_C & y_C \\ x_C & y_C \\ x_C & y_C \\ \dots & \dots \\ x_C & y_C \\ x_C & y_C \\ x_C & y_C \end{pmatrix} \tag{18}$$

An arbitrary arc is interpolated by using coordinate transformation.

3.3 Affine Transformation for spiral curve

As shown in Fig.7, an arbitrary spiral curve should be interpolated in the space coordinate system of XOYZ, and one of the axes deviates from other two axes in XOYZ space coordinate system. There is a plane called as “P”, which is perpendicular to the axis of spiral curve and going through the start point of the spiral curve. An arc is attainable by projecting the spiral curve onto the plane, then a coordinate system namely XOYZ, the center C of the plane P is the O, and the Z of which coincides with the axis of spiral curve, can be established, and the arc center is designated as its origin. Moreover, the positive direction of Z coincides with the spiral direction, and the X goes through the start point S.

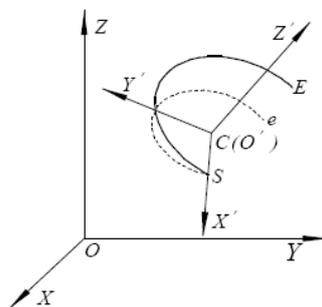


Fig.7 Affine transformation for spiral curve sketch.

The normal vectors of X axis, Y axis and Z axis are defined as i, j, k respectively, which in the XOYZ space coordinate system. The normal vectors of X axis, Y axis and Z axis that in the XOYZ space coordinate system are defined as i', j', k' respectively. Based on the theory of affine transformation [6], with the function of affine transformation, $O \rightarrow C(x_C, y_C, z_C), i \rightarrow i', j \rightarrow j', k \rightarrow k'$, the value

of the point C is $(0, 0, 0)$ in the $XOYZ$ space coordinate system, the affine transformation of the point E is as follow:

$$i' = \frac{OS}{|OS|} = [a_{11} \ a_{21} \ a_{31}] [i \ j \ k]';$$

$$k' = \cos\alpha \ i + \cos\beta \ j + \cos\gamma \ k = [a_{13} \ a_{23} \ a_{33}] [i \ j \ k]'; \quad j' = k' \times i' = [a_{12} \ a_{22} \ a_{32}] [i \ j \ k]'$$

The matrix formula is obtained as:

$$\begin{bmatrix} i' \\ j' \\ k' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix} = A^{-1} \begin{bmatrix} i \\ j \\ k \end{bmatrix} \tag{19}$$

Where the A is the matrix of affine transformation, and is also an orthogonal matrix. So, $A^{-1} = A^T$.

$$A = (A^T)^T = (A^{-1})^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{20}$$

Therefore:

$$\begin{bmatrix} i \\ j \\ k \end{bmatrix} = A \ A^{-1} \begin{bmatrix} i \\ j \\ k \end{bmatrix} = A \begin{bmatrix} i' \\ j' \\ k' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} i' \\ j' \\ k' \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} \tag{22}$$

The coordinate of the point E by transforming is:

$$\begin{bmatrix} x'_E \\ y'_E \\ z'_E \end{bmatrix} = A^{-1} \left(\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} - \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} \right) \tag{23}$$

The arbitrary spatial curve can be transformed into one or more basic segments in $XOYZ$ by using formula (23), and any arbitrary spatial curve is interpolated by using this theory of affine transformation.

3.4 Call for Basic Sub-modules of Interpolation

Taking a cylindrical spatial curve for example, the cylindrical spatial curves are projected onto the XOY plane, which are clockwise or counterclockwise arcs that are dotted arcs, these circular centers of which are placed in the origin of the plane XOY , and their start points are located in X . See Fig.8. The cylindrical spatial curve interpolation can be considered as an arc-interpolation in the plane XOY and Simultaneous Motion interpolation with Z -axis, along which the point from origin to positive direction.

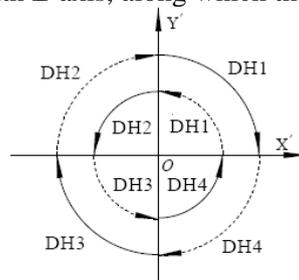
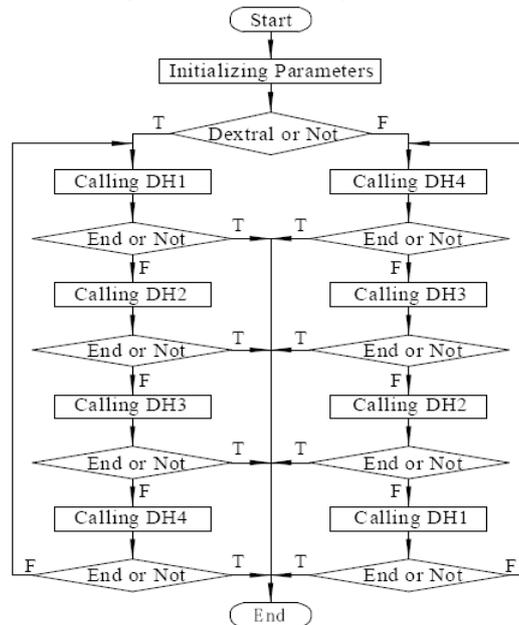


Fig.8 Interpolation subprograms for cylindrical Spiral curve in XOY .

The flowchart of interpolation for cylindrical spatial curve is exhibited in Fig.9. The parameters that include end point and feed speed and pitch and direction of turning are initialized. Next, with respect to the dextral cylindrical spatial curve, after distinguishing its direction of turning, the DH1 of the “Basic Sub-module of Interpolation” is used to implement interpolation periodically from the start point which is in X. Whether the interpolation is finished or not, it has been estimated by the *DH1* since the interpolation was terminated by itself if necessary. If the destination has been achieved,



the interpolation would be ended. However, if the destination has not been reached yet, while the *DH1* has been still exiting, the *DH2* will be called as main program. By this way, the *DH3* and the *DH4* are called in turn. The *DH1* is called again when the destination has not yet been reached but the curve of a lead has been finished and the current point returns to X. The case of levorotatory cylindrical spatial curve is similar to the dextral one, and the subprogram is called as follows: *LH4*→*LH3*→*LH2*→*LH1*→*LH4*→*LH3*→.....

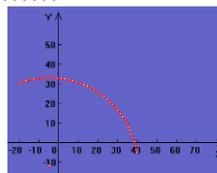
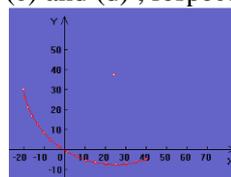


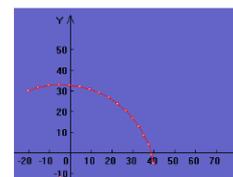
Fig.9 Interpolation subprograms for cylindrical Spiral curve in XOY.

4. Case Example

This period shows the examples of G02, G03 and different size step interpolation, which are shown in Fig.10 (a), (b), (c) and (d) , respectively.



(a) G02 interpolation



(b) G03 interpolation



```

%
M0010 G90 G94 G97 G21 G40
G49:
M0020 G92 X=100.000 Y200.000
Z0.000 T01;
M0030 G43 G00 Y 0.000 M01 S550
M03;
M0040 G42 G03 X0.057 Y100.0
Z329.850 R100.000 D01 M08 F40;
M0050 G01 X=120;
M0060 G49 G40 G00 X100.000
Y200.000 Z0.000 M09 M05;

```

Fig.10 Example of interpolation for arc.

The interpolation of spatial curve is developed based on Visual C++, the interface of example validation is shown in Fig.11.

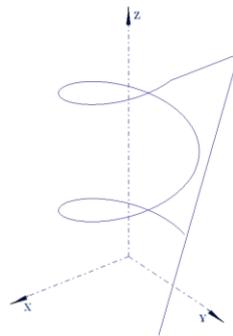


Fig.11 Example of interpolation for spatial curve.

5. Conclusion

In this study, the software platform of arc-interpolation and the spatial curve interpolation are promoted by Visual C++ through introduction of the principle of sampled-data Interpolation Algorithm and presetting of the flowchart of arc-interpolation. Adopting the theory of coordinate transformation, any arbitrary arc could be interpolated in the 2-D plane, and arbitrary spatial curve could be transformed into more basic segments by using affine transformation. The establishment of basic sub-modules of interpolation is based on arc-interpolation that belongs to sampled-data Interpolation Algorithm.

The thesis focuses on the development of the arc sampled-data interpolation algorithm software, and attempts to build the CNC virtual simulation system, which can replace the real machine debugging. With the innovation of interpolation algorithm for CNC system, it will shorten development cycle and save development cost.

Appendix

There are some Visual C++ programs that show as follows:

```

void CModelessDlg2::OnOk2()
{
    // TODO: Add your control notification handler code here

```

```

UpdateData (true); // user input
CClientDC dc(m_parent);
CPen BluePen(PS_SOLID, 1.5, RGB(0,255,0));
dc.SelectObject(&BluePen);
float X1,Y1;// circular center coordinate
float x1,y1,x2,y2; // two points coordinate
float r,d,l;// radius, the distance of two points and that to the arc center
int direct;//arc orientation, plus means clockwise
int cx,cy; // the parameters of arc enter
float ox,oy;
float dx,dy;
float sin0,cos0;
x1 = m_xa1;
y1 = m_ya1;
x2 = m_xb1;
y2 = m_yb1;
r = m_r;
if((x1==x2)&&(y1 == y2))
{
    dc.TextOut(600,380, "attention: input radius is too large! ");
    return;
}
if(fabsf(r)*2<fabsf(sqrt(pow((x1-x2),2)+pow((y1-y2),2))))
{
    dc.TextOut(600,380, "attention: input radius is too small! ");
    return;
}
direct = m_g - 1;
if (direct == 2) direct = -1;
if(r<0) direct = -direct;
d = sqrt(pow((x1-x2),2)+pow((y1-y2),2));
cos0 = fabsf(y1-y2)/d;
sin0 = fabsf(x1-x2)/d;
l = sqrt(r*r - d*d/4);
dx = l*cos0;
dy = l*sin0;
ox = (x1+x2)/2;
oy = (y1+y2)/2;
if(y1 == y2)
{
    cx = 0;
    if(x1>x2)
    {
        cy = direct;
    }
    else
    {
        cy = -direct;
    }
}
else

```

```

    {
        if((x1-x2)*(y1-y2)<0)
        {
            if(y1>y2)
            {
                cx = cy = -direct;
            }
            else
            {
                cx = cy = direct;
            }
        }
        else
        {
            if(y1>y2)
            {
                cx = -direct;
                cy = direct;
            }
            else
            {
                cx = direct;
                cy = -direct;
            }
        }
    }
}
... ..
}

```

Acknowledgments

This work was financially supported by A Key Project of Higher Education Natural Science Foundation of Anhui Province of China under Grant No. KJ2014A036 and KJ2015A383. We also acknowledge the financial support of “FOUNDATION OF WUHU TECHNOLOGY PLAN-NING” by Wuhu Work NC Productivity Promotion Center CO., LTD.

References:

- [1] Agrawal R K, Pratihari D K, Choudhury A R. Optimization of CNC is scallop free form surface machining using a genetic algorithm [J]. International Journal of Machine Tools and Manufacture, 2006, 46(7-8): 811-819.
- [2] Perfilieva I, Dubois D, Prade H, et al. Interpolation of fuzzy data: Analytical approach and overview Original Research Article [J]. Fuzzy Sets and Systems, 2012, 192(4): 134-158.
- [3] Erdos G, Muller M, Paul X, Parametric tool correction algorithm for 5-axis machining, Advances in Engineering Software, 2005, 36:654-663.
- [4] HU Jun, XIAO Lingjian, WANG Yahan, et al., An optimal feed rate model and solution algorithm for a high-speed machine of small line blocks with look-ahead, International Journal of Advanced Manufacturing Technology, 2006, 28(9): 930-935.
- [5] Fu-Yuan Luo, Yun-Fei Zhou, Juan Yin, A universal velocity profile generation approach for high-speed machining of small line segments with look-ahead[J], Springer, 2007, 505-518.

- [6] Moustris G, Tzafestas S G. Reducing a class of polygonal path tracking to straight line tracking via nonlinear strip-wise affine transformation [J]. *Mathematics and Computers in Simulation*, 2008, 79(2): 133-148.