Development of Canned Cycle for CNC Milling Machine

S.N Sheth[1], Prof. A.N.Rathour[2]

[1]Scholar, Department of mechanical Engineering, C.U SHAH College of Engineering & Technology

[2]Professor, Department of mechanical Engineering, C.U SHAH College of Engineering & Technology

Abstract— Despite the tremendous development in CNC programming facilities, linear and circular cuts parallel to the coordinate planes continue to be the standard motions of modern CNC machines. However, the increasing industrial demand for parts with intricate shapes cannot be satisfied with only these standard motions. The proportion of parts which are not covered by the standard CNC motions is certainly growing, due to the increasing industrial demand for intricate shapes. So for that we have decided to develop a new CANNED cycle with the help of MACRO (parametric) programming on Hypo cycloid, combined EP-hypo cycloid curve and last for Bezier surface also. we found that this canned cycle would be useful for different application such as cycloidal speed reducer, lobe pump rotor and surface milling without requiring external arrangement like CAD modeling, CNC interpolator etc.

Keywords— CANNED cycle, MACRO (parametric) programming, Hypo cycloid, EP-hypo cycloid, Bezier surface, cycloidal speed reducer, lobe pump rotor.

1. INTRODUCTION

In product development, stages such as conceptual design, prototype making, CAD model construction, tooling design, etc. are involved. Either forward engineering or reverse engineering can be employed. In general, in forward engineering, a 3D solid model of a product is first designed in a CAD platform and CAD information is obtained. Then corresponding tool-path information is generated and the product is produced using an appropriate manufacturing process, such as CNC machining. However, in reverse engineering, geometrical information is needed to be obtained from a physical shape directly and this information is converted into a computable format for other downstream processes. In most cases, a digitizing device is used to get the data and the output data is usually a set of scattered points (called a point cloud). Due to the characteristics of the digitizing device, the point cloud can be divided into two main types—a regular point cloud and an irregular point cloud. In the former, intervals between adjacent digitizing points are identical while they are not in the latter. No matter which type of point cloud is obtained, surface fitting techniques are usually applied and a surface model is constructed. Based on the surface model, CAM tool-path information is generated accordingly. This procedure is termed as an indirect machining process and the technologies for fitting surfaces onto a point cloud are essential. However, surface fitting process is usually time-consuming. The tool-path information can be achieved directly from the point cloud. But, this two technique (forward & reverse) required making much external arrangement and so more money and time. Why we are not going to use available internal facility? It means in CNC machine there is a facility of programming the tool path according to the required shape of job and for that we can use combination of linear, circular and curvature tool path. [1]

The raw data could, in a simple case, consist of coordinate for points which should be connected with straight lines or second degree curves fulfilling certain conditions of smoothness. Regardless of complexity, however, the translation should result in data for the complete path in the form of piecewise representation of mathematical curves. The curve data obtained from the translator is to be converted into small unit steps along the fixed axes. This process is also known as interpolation. [2]

2. TOOLS TO BE USED

Since the advent of the computer, the demand for complex shapes has been met by constantly upgrading the shape generating capabilities of CNC systems and by developing sophisticated CAD/CAM processors, capable of reducing complex geometries to long series of linear cuts. Thus, if a particular shape cannot be programmed directly with the standard CNC motions, it is first linearized with the help of a CAD/CAM system, which then encodes the result automatically into an executable NC program.[3]

The development and incorporation of tool path generators into CNC systems, based on efficient and accurate curve tracing methods, capable to satisfy the increasing industrial demand for machining complex shape parts is an important goal in the field of computer-aided manufacturing. Another frequent demand is met in the field of surface machining. A lot of sculptured surfaces as are the cases of molds, stamping dies, forging tools, rolling shapes, etc., are defined as revolved surfaces with free-form profiles. Despite the particularity in the definition and the design of these surfaces the available CAM systems deal with them as with free-form surfaces. That is, a sequence of straight lines is used to approximate the part surface and voluminous data describing them must be sent to the CNC machine. [4]

For boundaries formed at the intersection of higher degree or free-form surfaces, an accurate solution of the boundary machining problem has hitherto been considered to be beyond the power of traditional curve tracing methods Although related topics as are general surface intersection methods, problems raised on surface/ surface intersection and boundary representation methods have been extensively addressed by several authors. [5]

Parametric programming, mathematical calculations with do-loop subroutines, macro-capabilities and sophisticated canned cycles are among the strengths of the last CNC generation lessening the user's dependence on CAD/CAM. Despite this tremendous development in programming facilities, however, basic motions in a three axis CNC machine continue to be executed only by 2D or three-dimensional (3D) linear and 2D circular interpolators. Other types of motions implemented by approximating the desired path with straight-line segments are accompanied with the drawbacks of acceleration–deceleration cycles on the machine and consequently machining inaccuracies are raised with the machining time increasing substantially.

2.1 An Introduction to CANNED CYCLE

Canned cycles provide a programming method of a CNC machine to accomplish repetitive machining operations using theG/M code language. Essentially, canned cycles are a set of pre-programmed instructions permanently stored in the machine controller that automate many of the required repetitive tasks. Their use eliminates the need for many lines of programming, reduces the programming time and simplifies the whole programming process. [6] All CNC machining controls come with a set of helpful machining canned cycles. These canned cycles are executed or ''called up''n by entering a certain code together with any required variable information. Once canned cycle has been defined it remain active until cancelled. Drilling, counter-boring, peck drilling, pocket or slot machining are all examples of standard canned cycles. However, the standard canned cycles are limited in number and capability, being unable to accommodate the increasing needs of applications with complex geometries. Example of standard CANNED CYCLE Rough turning cycle: G71 Once the caned cycle has been defined it remains active until cancelled. In other word every time a block has axis movement programmed, the machining operation of the canned cycle is active also. Multiple function are defined as a series of function which allow a machining operation to be repeated along a given path. The programmer will select the type of machining which can be canned cycle or model subroutine. These functions must be defined every time they are used. The programming of canned cycle is readily available in the controller of the NC/CNC machines for from activating and geometry. However for some unknown geometry for which canned cycles are not available inparticular controller, some can be developed. Milling machine with particular set of different boundary under regular manner demands development of new canned cycle.

2.2 Macro programming

Macro programming provides a means of shortening code and doing repetitive tasks easily and quickly. All of your canned cyclesin a control are nothing but a macro. Macro is also extremely useful for families of parts. Repetitive operations can be made similar to using a sub program, but a macro will allow you to change conditions without having to edit multiple lines of code. Variables are used in place of coordinate numbers in the program. Variables are expressed in the program by a numerical value preceded by the pound sign #100, #500. Values can be input manually into a variable register Values can also be assigned via the NC program #510=1.5, #100=#100+1 Variables Can Only be Mathematical Quantities (numbers)

2.2.1 Program flow function

We need to understand some program flow (control) functions before we do our mathematics because we need some of these functions to quickly perform the mathematics. These are the three most commonly used: IF [compare1 {function} compare2]-GOTO[block]: The if-then statement is a conditional jump. If the statement is true then the GOTO command is executed and a program jump is performed. If the statement is false, then program flow continues with the next block. There are numerous functions available, the most common are: EQ - (==) – Equals NE - (<>) - Not Equal LT - (<) - Less than GT - (>) - Greater than Example: N40 IF [#500 EQ 0] GOTO 900 - If variable #500 equals 0 then jump to block 900. WHILE [compare1 {function} compare2] DO END: While the comparison is true, the blocks between DO and END are repeated, with the comparison checked each time it loops. Example: N10 WHILE [#530 LT 8] DO N40... N50.... N60..... N70 END N80....... So long as #530 is less than 8, blocks N40-N70 are executed repeatedly. When #530 is no longer less than 8, program execution jumps to block N80 and continues. GOTO {block or label}: This is an absolute jump to a different block. In the Siemens controls the commands are GOTOF and GOTOB depending on which way you want the control to search for the block. (F= Forward, B= Backwards) Siemens also supports labels, while Fanuc style does not. Mathematical Functions: There are quite a few mathematical functions available to us for macro programming. Some controls offer more extensive operation sets, but I'll stick with the Fanucese standard set for now. The standard operational order of equations for Fanucese is: First: Functions (Trig Functions, etc) Second: Multiplication, Division, AND Third: Addition, Subtraction, OR, XOR etc.

3. TOOL PATH GENERATION TECHNIQUE

Curves and surfaces are mathematically represented explicitly, implicitly or parametrically. Explicit representation of the form $y = f(x)$, although useful in many applications, are axis dependent, cannot adequately represent multiple-valued functions, and cannot be used where a constraint involves an infinite derivative. Hence, these are little used in computer graphics or computer aided design. Implicit representation of the form $f(x, y) = 0$ and $f(x, y, z) = 0$ for curves and surfaces, respectively, are capable of representing multiple-valued functions but are still axis dependent. However, these have a variety of uses in computer graphics and computer aided design.

3.1 Parametric Curves

Parametric curve representation of the form:

$$x=f(t); \qquad y=g(t); \qquad z=h(t); \tag{3.1}$$

Where t is a parameter, has extreme flexibility. These are axis independent, representing multiple-valued functions having infinite derivatives. These parametric curves have additional degrees of freedom compared to either explicit or implicit formulations. To see the later point, an explicit cubic equation is considered.

$$y = ax^3 + bx^2 + cx + d \tag{3.1.1}$$

Here, four degrees of freedom exist, one for each of the four constant coefficients a, b, c, d.

Rewriting this equation in parametric form,

$$x(t)= \alpha t^3 + \beta t^2 + \gamma t + \delta$$

$$y(t)=kt^3+lt^2+mt+n \tag{3.1.2}$$

Where $c_1 \leq t \leq c_2$

Here, eight degrees of freedom exist, one for each of the eight constant coefficients α , β , γ , δ , k ,l, m, n. Although not necessary, the parameter range is frequently normalized to $0 \leq t \leq 1$.

4. SELECTION OF CURVE

For instance trochoidal milling strategies efficient for industrial applications. A trochoidal tool path is defined as the combination of a uniform circular motion with a uniform linear motion. As a result, trajectory radius is continuous, which creates favorable milling conditions in terms of tool loads and kinematics. Furthermore, full-immersion milling configurations are avoided. Nevertheless, tool path length is much higher compared to standard tool paths such as zigzag because large portions are outside the material. Tool path interpolation has also a major influence on the process implementation. Thus, trochoidal tool paths are well adapted to complex milling cases, such as hard material roughing.[7] Based on geometric design of cycloidal speed reducer [8] and lobe pump rotor found that hypotrochodal curve is useful for the same.

4.1 EPTROCHOID—Definition and Parametric Representation

A curve traced by a point P fixed to a circle with radius r rolling along the outside of a larger, stationary circle with radius R at a constant rate without slipping, where the point P is at distance h from the center C of the exterior circle.

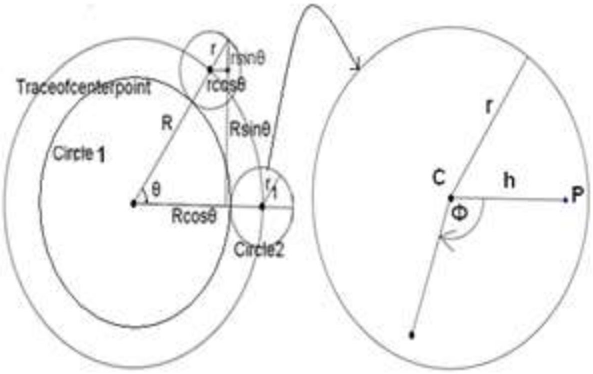


***Fig 4.1.1**: Epi-trochoid curve Representation*

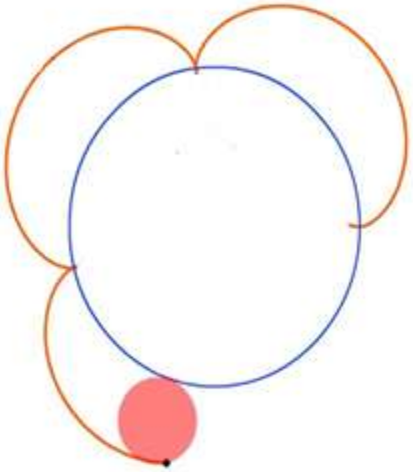


***Fig 4.1.2**: Epi-trochoid curve generation*

The parametric form of the curve is

$$X=r\cos\Theta + r\cos\Theta - h\cos\left(\frac{R+r}{r}\boldsymbol{\theta}\right) \tag{4.1.1}$$

$$Y=r\sin\Theta + r\sin\Theta - h\sin\left(\frac{R+r}{r}\boldsymbol{\theta}\right) \tag{4.1.2}$$

Where, R= Fix circle radius (large)
r = Rotating circle radius (small)

The derivation of the equation is given in Appendix

4.2 HYPOTROCHOID—Definition and Parametric Representation

A curve traced by a point P fixed to a circle with radius r rolling along the inside of a larger, stationary circle with radius R at a constant rate without slipping, where the point P is at distance h from the center C of the interior circle. The name hypotrochoid comes from the Greek word hypo, which means under, and the Latin word trochus, which means hoop.

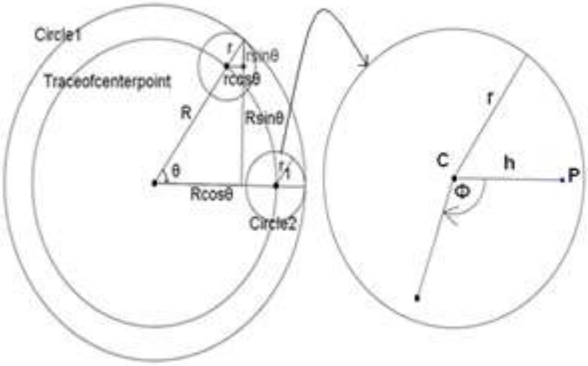


Fig 4.2.1: *Hypo-trochoid curve Representation*

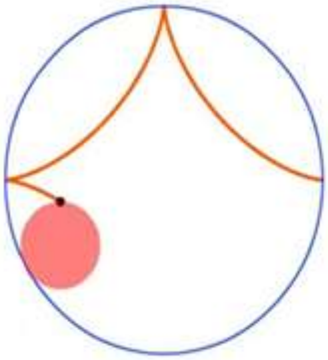


Fig 4.2.2: *Hypo-trochoid curve generation*

The parametric form of the curve is

$$X=r\cos\Theta - r\cos\Theta + h\cos\left\{\frac{R-r}{r}\boldsymbol{\theta}\right\} \qquad (4.2.1)$$

$$Y=r\sin\Theta - r\sin\Theta - h\sin\left\{\frac{R-r}{r}\boldsymbol{\theta}\right\} \qquad (4.2.2)$$

Where, R= Fix circle radius (large)
r = Rotating circle radius (small)
The derivation of the equation is same way as Epi-trochoid in Appendix

4.3 Combination of EPTROCHOID and HYPOTROCHOID

It is the curve traced by the two point P1 and P2 fixed to the circles with radii r rolling along the outside and inside respectively of a larger, stationary circle with radius R at a constant rate without slipping, where the point P1 and P2 is at distance h from the center C1 and C2 of the exterior and interior circles.

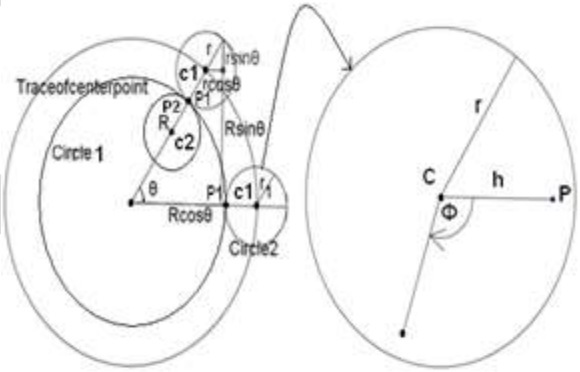


Fig 4.3.1: *Epi-trochoid curve Representation*

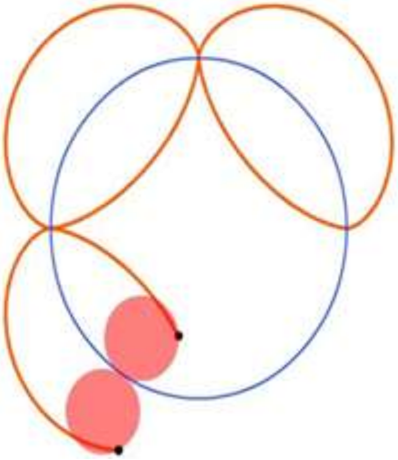


Fig 4.3.2: *Epi-trochoid curve generation*

4.4 Alternation of EPTROCHOID and HYPOTROCHOID

It is the curve traced by the two point P1 and P2 fixed to the circles with radii r rolling alternately for one complete revolution along the outside and inside respectively of a larger, stationary circle with radius R at a constant rate without slipping, where the point P1 and P2 is at distance h from the center C1 and C2 of the exterior and interior circles.

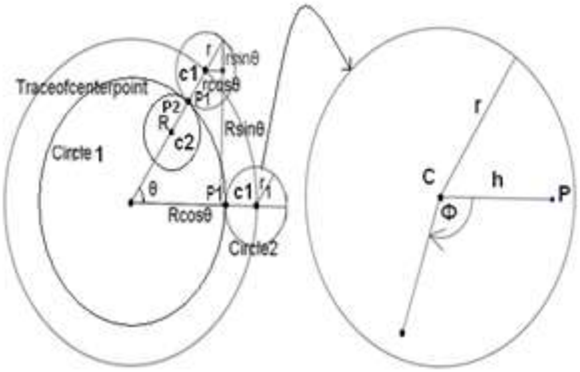


Fig 4.4.1: Alternate-Combined Epi-trochoid and Hypo-trochoid curve Representation

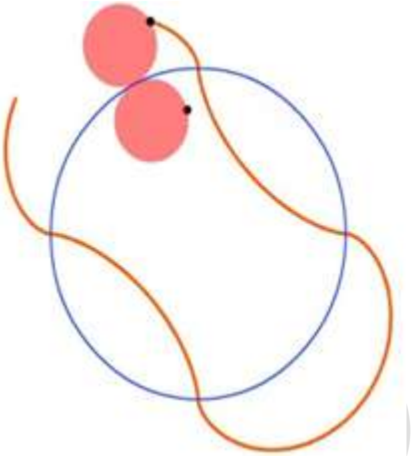


Fig 4.4.2: Alternate-Combined Epi-trochoid and Hypo-trochoid curve generation

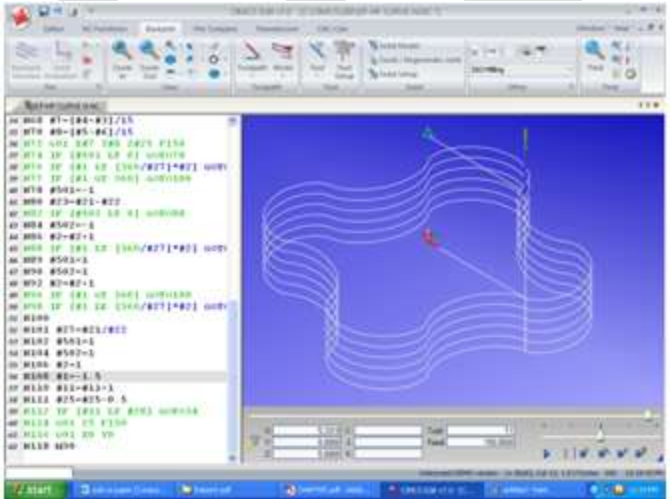
5. PART PROGRAMMING AND SIMULATION

Finally I found that alternate cycle of Combined EPTROCHOID and HYPOTROCHOID is useful for designing lobe pump rotor. So for that I have decided to develop a new canned cycle with help of macro programming.
Here **four teeth** lobe rotor cutting is programmed and simulated

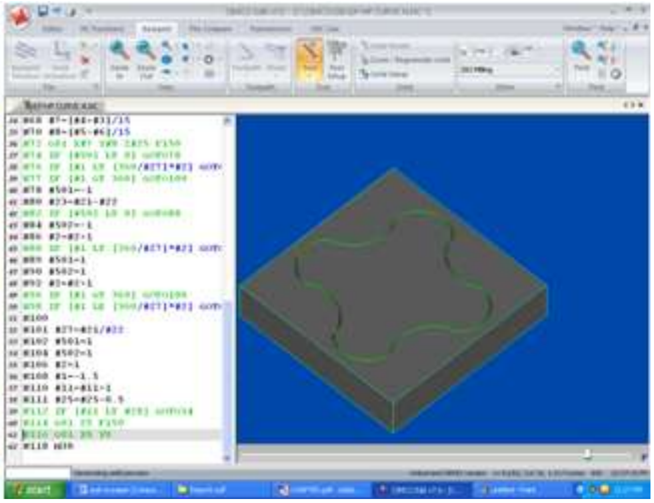
Macro–Program:

```
N10 G54 G90  M05
N12 G28 X0 Y0 Z0
N14 M06 T1
N16 G01 Z0 F(FEED)
N18 G41 D1
N20 #501=1
N22 #501=1
N23 #11=0
N25 #1=-1.5
N25 #2=1
N26 #21=(FIX CIRCLE DIAMETER)
N27 #22=(ROLLING CIRCLE DIAMETER)
N28 #24=(ARM LENGTH)
N29 #25=0
N29 #28=6
N30 S (SPINDAL SPEED) M03
N32 #27=#21/#22
N34 #23=#21+#22
N36 #26=#23/#22
N40 #1=#1+1
N42 #31=#26*#1
N44 #10=COS[#31]
N46 #3=#10*#24
N48 #13=COS[#1]
N50 #4=#13*#23
N52 #14=SIN[#1]
N54 #5=#14*#23
N56 #9=SIN[#31]
```

```
N58 #6=#9*#24
N60 IF [#501 LT 0] GOTO68
N62 #7=[#4-#3]
N64 #8=[#5-#6]
N66 IF [#501 GT 0] GOTO72
N68 #7=[#4+#3]
N70 #8=[#5-#6]
N72 G01 X#7 Y#8 Z#25 F150
N74 IF [#501 LT 0] GOTO78
N76 IF [#1 LT [360/#27]*#2] GOTO34
N77 IF [#1 GT 360] GOTO100
N78 #501=-1
N80 #23=#21-#22
N82 IF [#502 LT 0] GOTO88
N84 #502=-1
N86 #2=#2+1
N88 IF [#1 LT [360/#27]*#2] GOTO36
N89 #501=1
N90 #502=1
N92 #2=#2+1
N96 IF [#1 GT 360] GOTO100
N98 IF [#1 LE [360/#27]*#2] GOTO34
N100
N101 #27=#21/#22
N102 #501=1
N104 #502=1
N106 #2=1
N108 #1=-1.5
N110 #11=#11+1
N111 #25=#25-0.5
N112 IF [#11 LT #28] GOTO34
N114 G01 Z5 F150
N116 G01 X0 Y0
N118 M30
```



***5.1**: Four Teeth lobe pump rotor*



*Tool-Path **Fig 5.2**: Four Teeth lobe pump rotor*

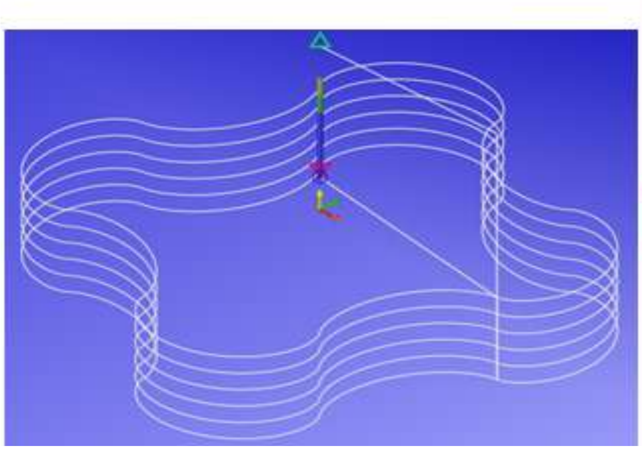


Fig 5.3: *Four Teeth lobe pump rotor Tool –Path*

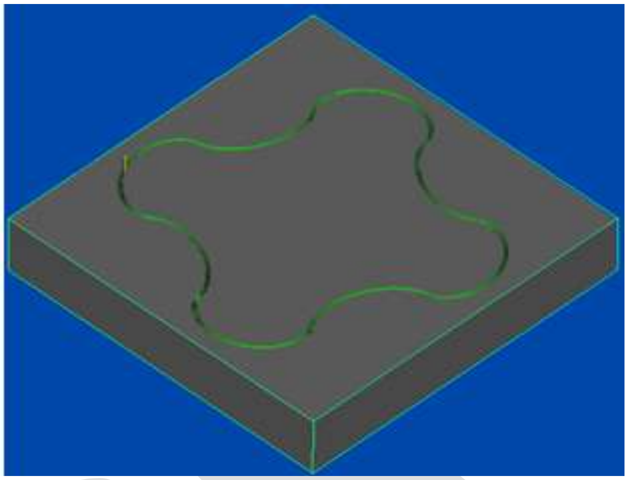


Fig 5.4: *Four Teeth lobe pump rotor*

Unassigned G code can be adopted to specify tool paths and associated federate functions. A block citing the preparatory function G contains one or more words that communicate information on the geometry and traversal rate of a curve. Many of the identifiers have established meanings within 'conventional' G code part programs e.g., angular dimensions about the coordinate axes for A, B, C; spindle speed for S; tool selection for T; federate for F; and secondary dimensions parallel to the coordinate axes for U, V, W.G blocks be modal—i.e., they specify values that will remain in effect until superseded by words of the same type in subsequent G blocks [9]

6. CONCLUSIONS

In the present work, a CANNED Cycle FOR lobe pump rotor has been developed with the parametric programming technique using Macro Programming. A tool path generation program has also been developed and Simulated in CIMCO Edit v7.0 simulation software. Developed canned cycle is useful to produce any number of teeth of lobe pump rotor with input of some key parameter of the curve.

- To make n teeth of lobe pump rotor take compression ratio (R/r)=2n
- Selection of radii is depends upon the size of rotor
- Selection of step value decides the smoothness and accuracy of the curve.

Developed CANNED Cycle can be called with G65 at any stage of part programming with input of some key parameter like radii R and r, speed, feed etc

7. APPENDIX

Derivation of Epi-trochoid parametric equation:
As the point C in Fig 4.1.1 travels through an angle Θ, than its x-coordinate is defined as
$x= (R\cos\Theta - r\cos\Theta)$ and y-coordinate is defined as
$y= (R\sin\Theta - r\sin\Theta)$.
The radius of the circle created by the center point is (R-r). As the small circle goes in a circular path from 0 to 2π, it travels in a counter-clockwise path around the inside of the large circle. However, the point *P* on the small circle rotates in a clockwise path around the center point C.
As the center rotates through an angle Θ, the point *P* rotates through an angle ϕ in the opposite direction. The point *P* travels in a circular path about the center of the small circle and therefore has the parametric equations of a circle.
However, since ϕ goes clockwise,
$x= h\ \mathrm{COS}\phi$ and
$y=-h\ \mathrm{SIN}\phi$.
Since the inner circle rolls along the inside of the stationary circle without slipping, the arc length $r\phi$ must be equal to the arc length $R\Theta$.
$r\phi=R\Theta$
$\phi=R\Theta/r$
However, since the point *P* rotates about the circle traced by the center of the small circle, which has radius (R-r), ϕ is equal to (R-r) × Θ/r. Therefore, the equations for a hypotrochoid are

$$X=r\cos\Theta + r\cos\Theta - h\cos\left\{\frac{R+r}{r}\boldsymbol{\theta}\right\}$$

$$Y=r\sin\Theta + r\sin\Theta - h\sin\left\{\frac{R+r}{r}\boldsymbol{\theta}\right\}$$

7.1 OTHER DEVELOPED CANNED CYCLE

7.1.1 HYPOTROCHOID POCKET:

The most popular method for machining a two-dimensional (2D) pocket is the contour-parallel offset method. Contour-parallel machining' is used to refer to pocketing with contour-parallel tool paths. However, we should note that the productivity of contour-parallel machining is mainly dependent on the tool path interval, because an increase in the tool-path interval brings a decrease in the total length of the tool paths. For die-cavity pocketing, contour-parallel machining is the most popular machining strategy. [10]

The generation of offset curves is a fundamental and well-known problem in CNC machining. The programmed path is the trajectory of the cutter center (CNC milling) or the center of the cutter's rounded tip (CNC turning), while the machined contour is the envelop of successive cutter positions. The programmed path is thus offset from the given contour by the cutter radius or the cutter's tip radius and we are faced with the problem of generating the offset as a real time trajectory. [11]

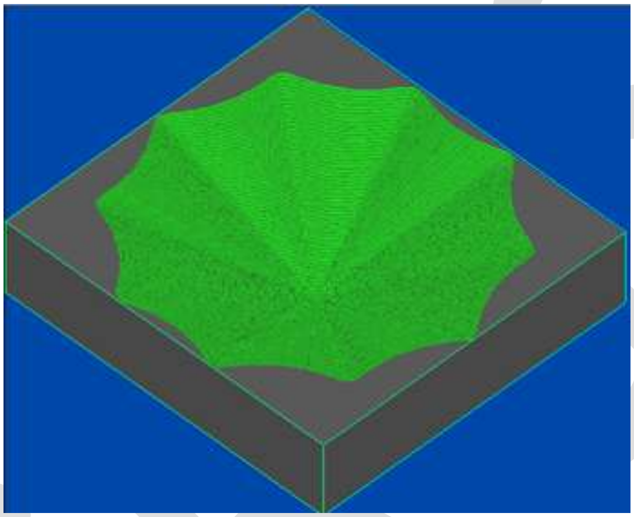


***Fig 7.1.1**: Hypo-trochoid Pocket*

7.1.2 BEZIER SURFACE

For generating a smooth movement in NC machining, parametric curve interpolator has been developed since 1990s. The input of parametric interpolator is a programmed tool path associated with an off-line or real-time scheduled federate, and from this a sequence of reference commands for the servo-controller can be outputted to coordinate the motion of each drive axis simultaneously. [12] But without requiring external arrangement like CAD modeling, CNC interpolator etc, we can generate required tool path. Parametric programming helps to generate Bezier surface. So for that I have developed canned cycle for the same.

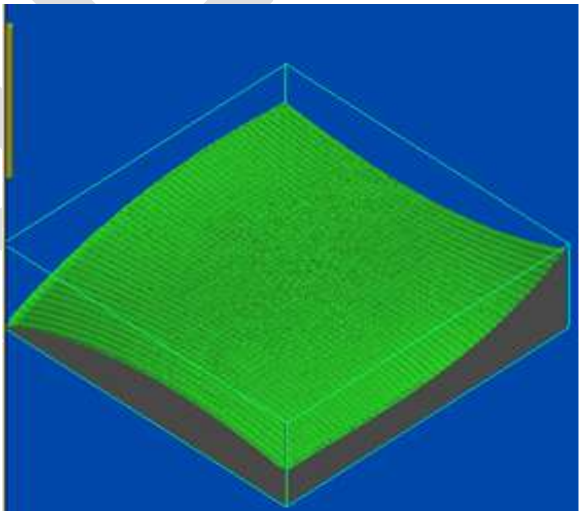


***Fig 7.1.2**: Bezier Surface*

REFERENCES

1. Biplab kanti haldar: "CNC Tool Path Generation For Free Form Surface Machining", Thesis submitted in Faculty of Engineering and Technology, Jadavpur University-2010.
2. PER E. DANIELSSON, MEMBER: "Incremental Curve Generation", IEEE TRANSACTIONS ON COMPUTERS, VOL. C-19, NO. 9, SEPTEMBER 1970
3. Sotiris L. Omirou.,: "Space curve interpolation for CNC machines", Journal of Materials Processing Technology 141 (2003) 343–350© 2003 Elsevier.
4. Sotiris L. Omirou, Antigoni K. Barouni: "Integration of new programming capabilities into a CNC milling system" Robotics and Computer-Integrated Manufacturing 21 (2005) 518–527, ©2004 Elsevier.
5. Sotiris L. Omirou: "A CNC interpolation algorithm for boundary machining" Robotics and Computer-Integrated Manufacturing 20 (2004) 255–264, © 2003 Elsevier.
6. Sotiris L. Omirou, Andreas C. Nearchou: "An epitrochoidal pocket—A new canned cycle for CNC milling machines", Robotics and Computer-Integrated Manufacturing 25 (2009) 73–80 © 2007 Elsevier.
7. Matthieu Rauch , EmmanuelDuc , Jean-YvesHascoet: "Improving trochoidal tool paths generation and implementation using Process constraints modeling" International Journal of Machine Tools & Manufacture 49 (2009) 375–383 © 2009 Elsevier.
8. Yii-Wen Hwang, Chiu-Fan Hsieh: "Geometry Design and Analysis for Trochoidal type Speed Reducers: with conjugate envelopes", No. 05-CSME-59, B.LC. Accession 2911,2006
9. Rida T. Farouki, Jairam Manjunathaiah, Guo-Feng Yuan: "G codes for the specification of Pythagorean-hodograph tool paths and associated feedrate functions on open-architecture CNC machines", International Journal of Machine Tools & Manufacture 39 (1999) 123–142, 1998 Elsevier.
10. S.C. Park, B.K. Choi: "Uncut free pocketing tool-paths generation using pair-wise offset algorithm", Computer-Aided Design 33 (2001) 739-746, © 2000 Elsevier.
11. Sotiris L. Omirou: "A locus tracing algorithm for cutter offsetting in CNC machining", Robotics and Computer-Integrated Manufacturing 20 (2004) 49–55 , ©2003 Elsevier.

Yuwen Sun, Yang Zhao, Yurong Bao, Dongming Guo: "A novel adaptive-federate interpolation method for NURBS tool path with drive constraints", International Journal of Machine Tools & Manufacture 77(2014)74–81, © 2014 Elsevier