

CONTOUR MACHINING OF SOME COMPLEX CURVES

Constantin MINCIU*, Andrei Petre SEBE**, Dumitru CATRINA*

* POLITEHNICA University of Bucharest, Romania

** Quadrivium International Retail Consultants SRL – Bucharest, Romania

Abstract. This paper presents an example of complex curves machining which covers the range of contour machining in 2 axes on numerically controlled machine tools. All utilized software was generated based on original Pascal, AutoLisp and TCL files, being implemented the studies and algorithms presented in [5].

Keywords: medial machining, contour milling, software, commands

1. Introduction

In order to do the numerical control of tool's trajectories was elaborated a software named ProCN. This software has the following functions:

- import the geometry of the workpiece from existent software;
- compute the machining trajectories – digitization;
- post-processing the trajectory to obtain the numerically control file;
- elaborate the documentation for machining;
- transfer the machining file to the numerically control equipment.

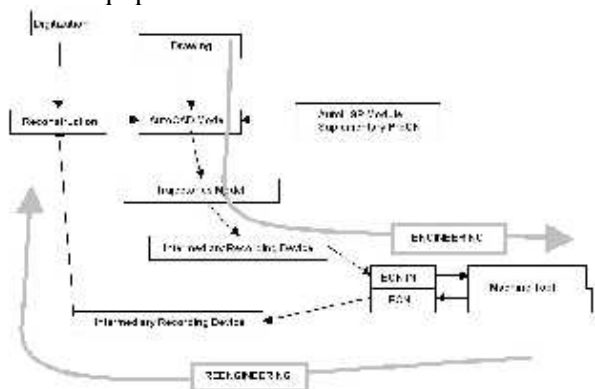


Figure 1. Principle scheme of ProCN software functions

2. ProCN programming

In the following, is presented an example of the use of this software at manufacturing of a shoe sole pattern.

Input the design data starts from a drawing (fig. 2) representing the contours of a pattern for the machining of shoe sole die. The data is input to the computer by scanning (minimum 300 dpi, black/white, line art) and it results a bitmap file.

The obtained computer image includes

imperfections due to the thermal noise of the scanner, spots on the paper, pencil imprecision etc. and it necessitates be processed and corrected with bitmap tools. The contours should be filled up in order to be expressed like vectors (figure 3).

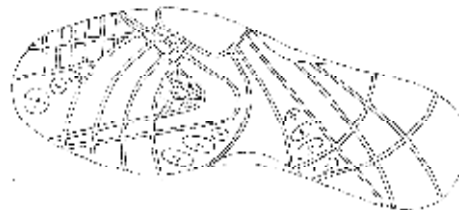


Figure 2. Shoe sole drawing from the designer



Figure 3. After scanning, contours are corrected and filled up

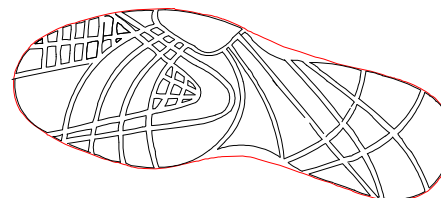


Figure 4. The vectorial drawing

In order to be used at the design of the pattern the image must be expressed by vectors, being obtained accurate edge contours (accuracy 0.1 mm) (fig. 4) and a file of vectors type (for example format EPS). Now the image (vectors) can be corrected and can be added, corrected or eliminated undesired contours (for example tablets

eventually added) with software like CorelDRAW (figure 4). Closed contours are intended to be obtained.

In order to be imported in AutoCAD, the image must be transformed in DXF format. A high accuracy of the curves description is intended (maximum error 0.05 mm).



Figure 5. Imported drawing, which has a lot of points



Figure 6. Drawing after FITARC command – over 70% less points

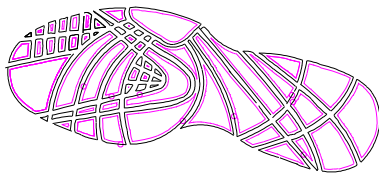


Figure 7. Trajectory of the machining tool obtained with the command TOFFSET

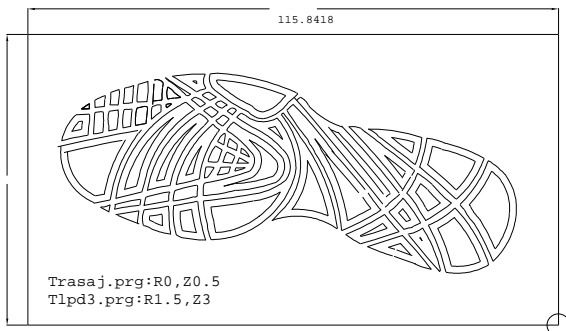


Figure 8. Document of the NC program



Figure 9. Result of a NC program simulation machining
AutoCAD software is opened and a new

drawing is created. By AutoCAD command DXFIN the desired image is imported. This image will be now consisting of polylines. The obtained drawing is saved, and it is reopened with the AutoCAD command OPEN. The un-useful layers and writing styles are purged with AutoCAD command PURGE. To save the drawing the AutoCAD command QSAVE is used (figure 5).

The software ProCN is loaded with the command (load "ProCN"). The complex curves which describe the contours are approximated by circle arches (with maximum error of 0.1 mm) using the ProCN command FITARC (figure 6).

Now, in layer 0 there is the approximation by circle arches of the selected contours for execution. These contours are transferred from layer 0 to layer FIT. Further, radius correction for the contours in layer FIT is done using the ProCN command TOFFSET (figure 7).

Now layer 0 contains the corrected contours, with desired radius. The origin of the reference coordinate system is set with AutoCAD command UCS. For the corrected contours the numerical control file is generated with ProCN command MAKE2DCN. After all these commands, in the current directory there is the numerical control file for the machining of the shoe sole pattern (it has approximately 500 programmed lines). Now is added the workpiece contour and the origin is set (point 0,0,0). This drawing is printed using the AutoCAD command PLOT (this will be the attached document to the disk containing the NC program) (figure 8). The drawing is saved by AutoCAD command QSAVE. AutoCAD software is quitted and DOS is used further.

Transfer to the file 1.720 of NC program is done with PUTPRG ProCN command. Now, in the image file disk RT11 there is the NC program. The floppy disk RT11 is inserted into drive A and the command WRITDISK is executed. On the floppy disk there is now the NC program.

The disk containing the NC program is inserted into the drive TU1. The workpiece and milling cutter are fixed accordingly to the attached document of the NC program. There are set: the milling cutter rotation sense, machining limits, origin of the reference system, feed speed, milling cutter rpm. The milling cutter is positioned manually in the starting point. NC program is started (figure 9).

The NC machining program was executed on STARRAG FIDIA milling machine.

3. Structure of ProCN program for 2D curves contouring

ProCN program adds supplementary functions to existent software, in order to make easier some steps of the numerical programming. It consists of DOS and AutoLISP sets of files.

DOS files add to DOS operating system some external commands, which allow the transfer of the information representing NC programs.

- READDISK – This program reads a floppy disk of 720k and transfers all its contents in a file named 1.720. The read disk must be formatted 720k in RT11 system or DOS and it must not have errors. Use: READDISK <enter>.

- WRITDISK – This program transfers the contents of a file named 1.720 onto a 720k floppy disk. The read disk must be formatted 720k in RT11 system or DOS and it must not have errors. Use: WRITDISK <enter>.

- GETPRG – This program extracts from a file named 1.720 all NC programs it finds and writes them into files named 1.prg, 2.prg etc. into the current directory of the current drive. Use: GETPRG 1.720 <enter>.

- PUTPRG – This program includes in a file named 1.720 from the current directory a specified NC program. In the current directory is mandatory to be found the file named ref.720. Use: PUTPRG [NC program name <enter>].

These routines are written in TurboPascal 6.0+Asambor language.

AutoLISP routines add a set of commands to AutoCAD software, in order to allow geometrical data processing.

- TOFFSET is the main command to set the technological offset. It takes a 2D selection and it generates an equidistant trajectory to the contours contained in this selection, at a prescribed radius of the milling cutter. For positive radius it gives a trajectory to the left related to the machining contour and for negative radius it gives a trajectory to the right related to the machining contour (considering the machining sense of the contour).

- DRAWFRAME is a command, which draws the minimum rectangle XY to include all the selected contours.

- INIPRO is a command to initialize implicit values of the global variables AFUZZ and DFUZZ. It is necessary to be executed before the other routines.

- AFUZZ is the variable which indicates to all ProCN routines the absolute minimum value of the center angle of a circle from which a circle arc may be considered as straight line (implicit value is 10^{-5}

radians).

- DFUZZ is the variable which indicates to all ProCN routines the absolute minimum value of the distance between two points from which these points may be considered identical (implicit value is 10^{-7} drawing units).

If necessary the last two variables can be modified using AutoLISP command SETQ.

- FITARC is a routine, which fits a set of circle and line segments with a prescribed error.

- MAKE2DCN is the NC processor for 2D contours. It takes a selection of machining contours and transforms them into a file containing NC commands.

In the following the internal ProCN commands are presented.

- TABFORM is a routine, which returns a value as string, formatted accordingly.

- MAKE2DCNSEGLIST is a routine, which takes a list of segments and writes it into a NC file.

- GETSEGINV is a routine, which returns a specified segment in inverse sense of machining.

- GETSEGOFF is a routine, which returns the offset of a segment by offset ro. If $ro > 0$ it returns a trajectory to the right of the segment and if $ro < 0$ it returns a trajectory to the left of the segment.

- DRAWSEG is a routine which draws the specified segment.

- SETSEG transforms LINE, ARC, VERTEX from AutoCAD format into an optimized entity (SEGMENT) having the following format: SEGMENT=(entn,spt,alf,bet,d,ept) where: *entn*= name of the entity (in the AutoCAD data base); *spt*=starting point of the segment (in current UCS coordinates); *ept*=ending point of the segment (in current UCS coordinates); *alf*=angle with OX axis of the chord of segment; *bet*=center angle of the segment curvature; *d*=length of the chord of segment.

- GETSENSSEGLIST is a routine that sets the machining sense of a segment list. For counter-clockwise sense it returns a positive number and for clockwise sense it returns a negative number.

- GETMAXMIN is a routine, which returns the diagonal of minimum rectangle, which enclose the argument point (extending its limits if necessary). This diagonal is a classic segment with the coordinates (xll, yll) (xur, yur).

- GETSEGFRAME is a routine, which returns the coordinates of the minimum contour, which enclose the argument segment.

- MAKESEGDR is a routine, which transforms a list of two points into a straight line segment.

- CUTSEG is a routine, which “cuts” a segment in two pieces and returns the list of these two resulted segments.
- DRAWSEGLIST is a routine, which gets as argument a segment list (assumed to be successive) and draws them as a polyline, returning the name of the created polyline.
- GETINTSEGLIST is a routine that returns a list containing the intersection points between a segment and a list of segments sl. There are not considered to be intersections the possible intersections of the segment with itself nor the segment starting and ending points.
- GETSEGLLEN is a routine, which determines the length of the segment.
- ORDSEGPTL is a routine, which gets a point list and a segment and returns a list of points in machining sense order.
- NEWITEM adds an item to the list ppl (if it is not yet contained in it) and returns the obtained list.
- CLEANLIST is a procedure to clean the segments having too small length and null segments from a list of segments.
- CHECKCONURLIST is a routine, which verifies the intersections of each contour in the argument segment list with another contour and returns a list containing only valid contours (offsets) if some segment of the contour intersects the main contour or if a circle on the middle point of segments intersects the main contour.
- PMEMBER is a function similar to the function member list of a point list. It returns the list starting with the first point at a smaller distance than DFUZZ from the argument point.
- GETCONURLIST is a routine, which gets a list of segments as argument, splits it in adjacent contours and returns this contour list. In the list, each element is a segment list.
- DRAWCONURLIST is a routine, which draws the argument contour list.
- CNEXT = circular next.
- GETSEGLISTOFF is a routine, which gets a segment, list sl as argument and calculates the segment list.
- GETSEGRAD is a routine, which returns the radius of the argument segment. If the center angle is zero, it forces to be not equal to zero by a very small value (AFUZZ).
- GETSEGCEN is a routine, which determines the center of a circle arc segment.
- ARCARC is a routine, which determines the intersection of two circle arcs.
- ARCLIN is a routine, which determines the

intersection of a circle arc with a straight line.

- LINLIN is a routine, which determines the intersection of two straight lines.
- DISTPTDR is a routine which determines the distance between one point and a straight line passing through the points x_1, y_1 and x_2, y_2 ; if the point is to the left of segment $d > 0$, otherwise $d < 0$.
- CHECKPT is a routine which returns the point if it belongs to the segment and it is null otherwise.
- GETSEGINT is a routine, which returns the intersection points of two segments, if they exist (otherwise returns zero for the illegal point). The intersection points should belong to the specified segments, in such order to describe the first segment in the machining sense.
- GETARC calculates the arc segments passing through the points (x_i, y_i) .
- GETPTSEGDIST is a routine, which calculates the distance between a point and its projection on a segment.
- GETERR determines the maximum error when approximating a segment list by circle arcs and straight lines.

4. Conclusions

1. This paper presents the NC contour machining of some complex curves.
2. NC programs were generated using original software, written in Pascal, AutoLISP and TCL languages, implementing the algorithms and studies presented in PhD thesis of Mr. A.P. Sebe.
3. The original software consists of DOS and AutoLisp set of files. These files were verified for the following: import of the workpiece geometry, machining trajectory calculation, NC program generation and its transfer to the NC equipment.
4. The NC programs were verified for the machining of a shoe sole pattern, used to manufacture the shoe sole dies, on STARRAG FIDIA NC milling machine.

References

1. Catrina, D. et al.: *Flexible systems for metal cutting*. Bren Publishing House, ISBN 973-648-231-6, București, 2003 (in Romanian)
2. Coquillart, C.: *Computing offsets of B-spline curves*. Computer-Aided Design, 19(6), 1987, 305-9
3. Ganea, C.C.: *Manufacturing technology analysis of spatial surfaces*. Tech. Univ. of Cluj Napoca, 2003 (in Romanian)
4. Schulz, H., Hockl, C.: *High-speed milling of dies and moulds-cutting conditions and technology*. 1995, 1 Annals of CIRP 44(1)1 35-8
5. Sebe, A.P.: *Researches regarding complex surfaces manufacturing on CNC machine-tools*. PhD Thesis, “POLITEHNICA” University of Bucharest, 2004, Romania (in Romanian)