

Proiectarea de postprocesoare pentru mașini cu comandă numerică

**Cercetător științific, Virgiliu Streian
Universitatea “Tibiscus” din Timișoara**

ABSTRACT. The purpose of this article is to become familiar with design of numerical control (NC) part programming of the computer aided manufacturing (CAM) process. This part consists in design of the CAM programs (postprocessors). The postprocessors are programs that convert a drawing, created by AutoCAD2000 by example, into G-code instructions which are used by G-code interpreter programs that in turn run the machine's controller. Conversion of the drawing to a g-code file is an important step. The conversion must be accurate in order to enable the CNC machine part production with minimum human intervention.

1 Descrierea unui sistem CAD/CAM

Proiectarea Asistată de Calculator - Computer aided design (CAD) și Lansarea în Fabricație Asistată de Calculator - Computer Aided Manufacturing (CAM) sunt două etape care se ocupă de automatizarea funcțiilor ingineresti legate de proiectarea și lansarea în fabricație a unui produs. CAD-ul include probleme legate de concepție, proiectare, modelare, simulare, analiză, evaluare și în final documentația pentru realizare produs, incluzând și planșe de execuție. Modelul geometric format pe durata procesului CAD este intrarea de bază pentru procesul CAM. AutoCAD, program realizat de firma AutoDesk Inc. din California SUA, este unul din numeroasele programe CAD utilizate în crearea de modele geometrice. CAM-ul include partea de programare utilizate în planificare și controlul numeric (CN) a mașinilor cu comandă numerică (CNC machine).

În sisteme CAD/CAM fazele de proiectare și lansare în fabricație sunt integrate în același sistem de calcul. Ieșirea dintr-un sistem CAD,

specificațiile de realizare a unui produs, sunt automat convertite către un sistem CAM de realizare a produsului prin intermediul unui program g-code numit în continuare postprocesor sau program CNC. Postprocesorul este descărcat într-o mașina cu comandă numerică prin intermediul unei interfețe de comunicare și astfel realizarea fizică a unui produs este implementată.

2 Privire generală și explicații despre mașini cu comanda numerică (CNC machine)

Controlul mașinilor cu comandă numerică prin calculator (CNC) a dat posibilitatea producerii unei largi și complexe game de mașini unelte. Principalele părți a unui sistem CNC sunt: (1)Un calculator personal(PC), (2)Programe care permit utilizatorului să proiecteze (ex. AutoCAD), (3)Programe care convertesc proiectul în instrucțiuni, instrucțiuni utilizate de programele de interpretoare G code, (4)Programe care interpretează instrucțiunile G code și comandă mașina uneltă.

PC-ul este destinat rulării unui software specializat care are drept scop final emiterea către mașina uneltă a unor semnale constând din instrucțiunile G code.

Prima componentă a acestui software constă din programe de proiectare numite generic programe de “Proiectare Asistată de Calculator-Computer Aided Design (CAD)”. Programul CAD la care ne vom referi în continuare va fi AutoCAD.

A doua componentă a acestui software constă din programe de lansare în fabricație a produsului proiectat, programe numite generic Computer Aided Manufacturing (CAM). Un program CAM alias un postprocesor, are drept scop generarea de instrucțiuni G code alias RS-274D standard, pentru o anumită mașină uneltă. Există diferite tipuri de formate g-code formate depinzând de tipul mașinii unelte utilizate.

RS-274D, sau altfel numit G-Code, este un standard recomandat pentru mașinile cu comandă numerică. Acest standard reprezintă un element de bază în scrierea programelor pentru comanda mașinilor unelte.

Programarea mașinilor cu comandă numerică (CNC programming) constă din planificarea și documentarea secvențelor, una câte una, pe care mașina uneltă le execută. Planificarea secvențelor presupune cunoașterea foarte bine a mașinii unelte respective, precum și noțiuni avansate de matematică, în special de geometrie și trigonometrie.

Limbaajul standard de programare a mașinilor unelte cu comandă numerică (RS274D alias G-code) conține instrucțiuni care specifică o serie

ordine adresate mașinii unelte. De regulă un program CNC este compus din următoarele părți: (1) o secțiune de inițializare, care pregătește de lucru mașina, (2) o secțiune de lucru efectiv a mașinii, (3) o secțiune de terminare a lucrului.

Următoarea secvență de program G-code exemplifică cele afirmate mai sus - Nxxxx este numărul liniei, Gn este un G code, Mn este un M code, Q un increment de alimentare, iar Fn este rata de alimentare dorită:

% header section – (1)une de inițializare

N0000 G91 X0 Y0 F200.000

N0005 G65 H01 P32 Q000.200

% (2)secțiune de lucru efectiv a mașinii

% pen down

N0010 M12

N0015 G01 X000.000 Y-85.024

N0020 G01 X118.340 Y000.000

N0025 G01 X000.000 Y085.024

N0026 G01 X-118.340 Y000.000

% pen up

N0030 M13

% moving to next entity

N0035 G00 X-97.457 Y-78.068

% pen down

N0040 M12

N0045 G01 X059.557 Y017.005

N0050 G03 X-07.734 Y035.556 I-05.006 J017.530

N0055 G02 X-30.939 Y017.777 I-04.286 J028.352

N0060 G01 X-20.884 Y-70.338

% (3)secțiune de terminare

% Pen up at end of program

N0065 M13

N0070 M02

De regulă G code-urile indică o mișcare în timp ce M code-urile sunt utilizate pentru controlul unor funcții de I/O a mașinii.

În listele de mai jos sunt explicitate câteva din codurile G/M recunoscute.

G Codes List

G code	Function and Description	Example
--------	--------------------------	---------

G00	Rapid Positioning. Using a G00 in your code is equivalent to saying "go rapidly to point xxxx yyyy". This code causes motion to occur at the maximum traverse rate	N100 G00 X10.00 Y5.00 - go rapidly to point X10.00 Y5.00 N100 G00 Z1.5 - move spindle above obstacles
G01	Linear Interpolation. G01 causes the machine to travel in a straight line with the benefit of a programmed feed rate (using "F" and the desired feedrate). This is used for actual machining and contouring.	N120 G01 Z0.1 F6.0 - move the tool down to Z=0.1 at a rate of 6 inches/minute N130 G01 Z-.125 F3.0 - move tool into the work piece at 3 inches/minute N140 G01 X2.5 F8.0 - move the table, so that the spindle travels to X=2.5 at a rate of 8 inches/minute
G02 G03	Circular/Helical Interpolation (CW/CCW). G02/G03 causes clockwise / counterclockwise circular motion to be generated at a specified feed rate (F). When coding circular moves, you must specify where the machine must go and where the center of the arc is in either of two ways: By specifying the center of the arc with J words, and I or giving the radius as an R word. I is the incremental distance from the X starting point to the X coordinate of the center of the arc. J is the incremental distance from the Y starting point to the Y coordinate of the center of the arc.	N100 G01 X0.0 Y1.0 F20.0 - go to X0.0, Y1.0 at a feed rate of 20 inches/minute N200 G02 X1.0 Y0.0 I0.0 J-1.0 - go in an CW arc from X0.0, Y1.0 to X1.0 Y0.0, with the center of the arc at X0.0, Y0.0 N300 G01 X0.0 Y1.0 F20.0 - go to X1.0, Y0.0 at a feed rate of 20 inches/minute N400 G02 X1.0 Y0.0 R1.0 - go in an CW arc from X0.0, Y1.0 to X1.0 Y0.0, with a radius of R=1.0
G20	Inch system selection.	
G21	Millimeter system selection.	

G90	Absolute Command. Use the absolute coordinates in G00, G01, etc. g codes	
G91	Increment Command. Use the incremental coordinates in G00, G01, and etc. g codes.	

M codes List

M code	Function
M00	Stop and wait for start button to restart
M02	End of Program
M03	Stop and Go to Console Mode
M11	Spindle on clockwise
M12	Spindle on counterclockwise
M13	Spindle off
M21	Coolant on
M22	Coolant off
M30	Output on
M31	Output off
M98	Call subprogram
M99	End of subprogram, return

3 Proiectarea unui postprocesor (CNC programs) utilizând limbajul AutoLISP din AutoCAD

AutoLISP, dialect pentru Common Lisp, este un limbaj de programare de nivel înalt ce reprezintă o facilitare standard a AutoCAD-ului. Prin AutoLISP se poate avea acces la informațiile din baza de date geometrică a entităților grafice. Această facilitare se poate folosi la proiectarea unui postprocesor (CNC programs).

De exemplu putem considera conturul poligonal 2D din Figura 1, creat cu comanda AutoCAD, PLINE. c

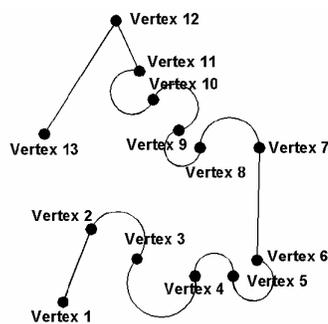


Figura 1.

Vertexurile 1, 6 și 12 stătează a segment de linie în timp ce vertexurile 2, 3, 4, 5, 7, 8, 9 și 10 stătează un arc de cerc. Tabelul de mai jos descrie caracteristicile acestor vertexuri:

Verte x	Xstart	Ystart	Radius of Arc (R<0 -- CW) (R>0 -- CCW)	Xcenter of Arc	Ycenter of Arc
1	84.6485	68.051			
2	97.332	100.561	-12.4125	108.896	96.0498
3	117.736	87.3367	12.4125	108.896	96.0498
4	143.654,	79.6224	-8.72067	152.202	81.3514
5	160.749	79.6224	8.72067	152.202	81.3514
6	171.227	86.7857			
7	172.33	136.929	13.2381	159.095	137.22
8	145.86	136.929	-8.00818	137.854	136.752
9	136.485	144.643	8.00818	137.854	136.752
10	124.905	158.418	-9.88147	115.604	161.755
11	118.839	171.092			
12	108.361	193.684			
13	76.3767,	142.99			

Lista de asocieri a acestei polilinii obținută cu secvența AutoLISP: (entget (nth 0 (entsel "Please choose a 2D polyline object:")) este arătată mai jos:

```
((-1 . <Entity name: 19e9d70>) (0 . "LWPOLYLINE") (330 . <Entity name: 19e9cf8>) (5 . "2E") (100 . "AcDbEntity") (67 . 0) (410 . "Model") (8 . "cote") (100 . "AcDbPolyline") (90 . 13) (70 . 0) (43 . 0.3) (38 . 0.0) (39 . 0.0) (10 84.6485 68.051) (40 . 0.3) (41 . 0.3) (42 . 0.0) (10 97.332 100.561) (40 . 0.3) (41 . 0.3) (42 . -1.22691) (10 117.736 87.3367) (40 . 0.3) (41 . 0.3) (42 . 1.6646) (10 143.654 79.6224) (40 . 0.3) (41 . 0.3) (42 . -1.22252) (10 160.749 79.6224) (40 . 0.3) (41 . 0.3) (42 . 2.46234) (10 171.227 86.7857) (40 . 0.3) (41 . 0.3) (42 . 0.0) (10 172.33 136.929) (40 . 0.3) (41 . 0.3) (42 . 1.02224) (10 145.86 136.929) (40 . 0.3) (41 . 0.3) (42 . -2.17968) (10 136.485 144.643) (40 . 0.3) (41 . 0.3) (42 . 1.73951) (10 124.905 158.418) (40 . 0.3) (41 . 0.3) (42 . -2.39575) (10 118.839 171.092) (40 . 0.3) (41 . 0.3) (42 . 0.0) (10 108.361 193.684) (40 . 0.3) (41 . 0.3) (42 . 0.0) (10 76.3767 142.99) (40 . 0.3) (41 . 0.3) (42 . 0.0) (210 0.0 0.0 1.0))
```

Fișierul G code creat de un postprocesor (program CNC) AutoLISP pentru o mașină unealtă de tăiere cu flacără, de exemplu, trebuie să prevadă următoarele faze: (1)extragerea listei de asocieri a entității polilinie 2D, (2) extragerea coordonatelor X, Y pentru fiecare vertex din lista de asocieri a polilinie (cod DXF 10), (3)testarea dacă vertexul startează un segment de linie sau un arc de cerc (cod DXF 42), (4)generarea pentru fiecare vertex a unei instrucțiuni, (5)scrierea, pentru fiecare vertex, a unei linii G-code într-un fișier disc

Un exemplu de postprocesor numit G_CODE (nume funcție main din fișierul CNC.LSP), scris în AutoLISP este arătat mai jos. Aplicația (două funcții lisp) citește lista de asocieri a poliliniei 2D atătată în Figura 1 și generează instrucțiuni G code instructions, care sunt salvate în fișierul GCODE.CNC.

```
;;
;;
;; File CNC.LSP
;;
;; AutoCAD2000 code
;; This code is only for training purposes
;; Author: V. Streian
;;
(defun CNC (PAL I)
  (if (/= PAL (list)) ; PAL – two-dimensional polyline association-list not
    empty?
    ; Then
    (progn
      (if (= (nth 0 (setq V (nth 0 PAL))) 10)
        ; then - write records into gcode file
        ; Nxxxx Ggg Xxxxx Yyyyy Rrrrr
        ; Ex.
        ; N1010 G90 -- absolute Command (absolute mode)
        ; N1010 G21 -- millimeter system selection
        ; N1010 G00 X100.00 Y100.00 --- go rapidly to point xxxx yyyy
        ; N1010 G01 X100.00 Y100.00 --- moves to point xxxx yyyy
        ; N1010 G03 X100.00 Y100.00 R50 --- go in an CCW arc from current
        point to xxxx yyyy (incremental point) with a R50 radius
        ; Only g-code G90, G21, G00, G01 and G03 are generated
        (progn
          (setq XY (cdr V)) ; extract X,Y of vertex
```

```
(setq X (nth 0 XY)) ; extract X of vertex
(setq Y (nth 1 XY)) ; extract Y of vertex
(if (= I 1000)
; then – only for first record
  (progn
    (princ
      (strcat
        "N"
        (itoa (setq I (+ I 10))))
      ) gcode) ; Nxxxx - line number
    (princ " " gcode) ; blank
    (princ "G21" gcode) ; G21 -- millimeter system
  selection
    (princ " " gcode) ; blank
    (princ "G90" gcode) ; G90 -- Absolute (coordinates)
  Command
    (princ "\n" gcode) ; end record -- a newline
;
  (princ
    (strcat
      "N"
      (itoa (setq I (+ I 10)))) ; Increments the counter by ten
    ) gcode) ; Nxxxx - line number
    (princ " " gcode) ; blank
    (princ "G00" gcode) ; G00 -- go rapidly to point X, Y
  (absolute mode)
    (princ " " gcode) ; blank
    (princ (strcat "X" (rtos X)) gcode) ; Xxxxx
    (princ " " gcode) ; blank
    (princ (strcat "Y" (rtos Y)) gcode) ; Yyyyy
    (princ "\n" gcode) ; end record -- a newline
  )
) ; end if
(princ
  (strcat
    "N"
    (itoa (setq I (+ I 10))))
  ) gcode) ; Nxxxx - line number
(princ " " gcode) ; blank
(setq TG (cdr (assoc 42 PAL))) ;extract tangent of bulge
```

```
(if (= TG 0) ; testing if vertex start a segment line or start an circular arc
; then – vertex start a line
  (progn
    (princ "G01" gcode) ; G01 -- moves to point xxxx yyyy
    (absolute mode)
    (princ " " gcode) ; blank
    (princ (strcat "X" (rtos X)) gcode) ; Xxxxx
    (princ " " gcode) ; blank
    (princ (strcat "Y" (rtos Y)) gcode) ; Yyyyy
  )
; Else – vertex start an arc
  (progn
    (setq ATGR (* 4.0 (atan TG))) ; AngleArc = 4 * bulge arc - radian
    angle
    (setq ATG
      (cvunit ATGR "radian" "degree") ;convert radian to degree angle
    )
    (setq C (osnap XY "_cen")); center of arc
    (setq R (distance XY C)) ; radius of arc
    (setq E (polar C ATGR R)) ; endpoint of arc
    (setq XI (- (nth 0 E) X)) ; incremental X of endpoint
    (setq YI (- (nth 1 E) Y)) ; incremental Y of endpoint
    (if (< TG 0) (setq R (* -1 R))) ; r=-r for clockwise
    (princ "G03" gcode) ; G03 -- go in an CCW arc from current point to
    xxxx yyyy
    ; (Incremental point) with a Rrrrr radius
    (princ " " gcode) ; blank
    (princ (strcat "X" (rtos XI)) gcode) ; Xxxxx
    (princ " " gcode) ; blank
    (princ (strcat "Y" (rtos YI)) gcode) ; Yyyyy
    (princ " " gcode) ; blank
    (princ (strcat "R" (rtos R)) gcode) ; Rrrrr
  )
) ; end if
(princ "\n" gcode) ; end record – a newline
) ; end prog
) ; end if
(setq PAL (cdr PAL))
(CNC PAL I) ; recursive call
) ; end prog
```

```
; else
  (progn
    (princ) ; Exit quietly
  )
); end if
) ; end CNC function
;;-----
;;
;;
;; Main function
;;
(defun C:G_Code ()
  (setq PAL ; retrieves the two-dimensional polyline association-list
    (entget (nth 0 (entsel "Please choose a 2D polyline object:"))))
  )
  (setq gcode (open "GCODE.CNC" "w"))
  (CNC PAL 1000)
  (close gcode)
  (princ "\nFile GCODE.CNC created")
  (princ)
) ; end G_code function
;;
;; End of CNC.LSP file
;;
```

Fișierul rezultat GCODE.CNC, arătat mai jos, conține o secvență de comenzi adresate unei mașini unelte de tăiere cu flacără, ce execută tăierea a conturului definit în Figura 1

```
% Fișierul rezultat GCODE.CNC
N1010 G21 G90
N1020 G00 X84.6485 Y68.051
N1030 G01 X84.6485 Y68.051
N1040 G03 X0.161 Y0.3927 R-12.4125
N1050 G03 X-15.7777 Y-1.5797 R12.4125
N1060 G03 X0.5124 Y5.1181 R-8.7207
N1070 G03 X-8.3059 Y-6.9884 R8.7207
N1080 G01 X171.2271 Y86.7857
N1090 G03 X-26.4603 Y-0.291 R13.2381
N1100 G03 X-9.2009 Y7.7425 R-8.0082
```

N1110 G03 X-2.5839 Y-14.8553 R8.0082
N1120 G03 X-9.4087 Y13.2175 R-9.8815
N1130 G01 X118.8388 Y171.0918
N1140 G01 X108.3611 Y193.6837
N1150 G01 X76.3767 Y142.9898

Bibliografie

- [Dog88] **Dogaru, Dorian** - *Metode noi în proiectare, Elemente de grafică tridimensională*, Editura științifică și enciclopedică, București, 1988.
- [Helxx] **Hello CadFANS** – *revistă românească de proiectare asistată*.
- [HBP02] **Harrington, Burchard, Pitzer** – *AutoCAD 2002*, Editura Teora, București, 2002.
- [Sim03] **Simion, Ionel** – *AutoCAD 2000 – Aplicații*, Editura Teora, București, 2003.
- [Sta93] **Stăncescu, Constantin** – *AutoCAD, manual de inițiere*, Editura FAST 2000, București, 1993
- [Sta96] **Stăncescu, Constantin** – *AutoLISP, manual de programare*, Editura FAST 2000, București, 1996.

Tibiscus