STEP-NC Interpolator for General 2D and 3D Parametric Curve

Dzullijah Ibrahim[*], *Yusli Yaakob, Norasikin Hussin*
Faculty of Mechanical Engineering,
Universiti Teknologi MARA, Cawangan Pulau Pinang,
Kampus Permatang Pauh, Pulau Pinang, Malaysia

Zahurin Samad
School of Mechanical Engineering,
Universiti Sains Malaysia, Pulau Pinang, Malaysia

[*]*dzullija@ppinang.uitm.edu.my*

ABSTRACT

The development of Open Architecture (PC-based) Numerical Control (OAPC-NC) Interpolator has enhanced the possibilities of data communication between the systems. In specialty domain, such as design, manufacturing, and machining, the STEP AP238 (STEP-NC) is developed. A STEP-NC interface to OAPC-NC interpolator using STEP-NC tool path data is proposed and developed. A hierarchical-based algorithm is used to extract the tool path data from STEP-NC tool path file of a product model. The generated output of the interpolated data is computer simulated for 2D circular-arc paths and 3D linear paths to verify the validity of the interpolator input generated by the proposed interface.

Keywords: *STEP-NC, open-architecture controller, parametric tool path.*

Introduction

Standard for data exchange formats such as Data eXchange Format (DXF), Initial Graphic Exchange Standard (IGES) and Product Description Exchange for Standard (PDES) have been developed over the years to overcome data communication problems between CAD/CAM systems. However, they are successful only in some limited applications, such as

ISSN 1823- 5514, eISSN 2550-164X
Received for review: 2017-04-30
Accepted for publication: 2017-05-06
Published: 2017-09-15

representing design and topology information in a platform independent format [1]. Therefore, in 1994, the International Standard Organization (ISO) adopted Standard for the Exchange of Product Model Data (STEP) as ISO 10303 to support product data exchange, independent of proprietary vendor CAD/CAM or other systems format. Now, nearly every major CAD/CAM system contains a module to read and write data defined by one of the STEP Application Protocols (AP's). Most commonly implemented protocol is called STEP AP-203 [2], which is used to exchange data describing designs represented as solid model parts and assemblies of solid models. By implementing STEP within CAD/CAM systems, the data exchange barrier is removed.

Yet, communication barriers exist between Computer Aided Design/Computer Aided Manufacturing (CAD/CAM) and Computer Numerical Control (CNC) systems due to G-codes weaknesses. These have been mentioned in literature, such as lacking power in execution and programming environment [3], unable to describe higher-level information about the manufacturing process [4], and most importantly, its inability to directly used CAD data [5]. Because of these reasons and other possible limitations of the standard, controller and machine tools manufacturers have introduced their own extensions [6] that are not covered in the limited scope of ISO 6983 [7], standard for G-codes implementation. Often happens that even controls from the same control manufacturer have an incompatible set of commands. This has made the CNC programs more proprietary and hardware dependent.

There are on-going international efforts which started a decade ago to replace the ISO 6983 M and G code standard with a modern associative language that connects the CAD design data used to determine the machine requirements for an operation with the CAM process data that solves those requirement [8]. STEP AP238 [9], commonly known as STEP-NC is a new model of data transfer developed for that purpose [10]. It uses the modern geometric constructs in the previous STEP (Standard for the Exchange of Product Data) neutral data standards to define device independent tool paths, and CAM independent volume removal [5]. The key feature of STEP-NC is the intelligent software in the machine controller will determine how to execute the task encoded in a STEP-NC file, based on the capability of the machine and the cutting tools with which it is equipped [11].

The concept of open architecture controller (OAC) of machine tools surfaced in the early nineties, helps to address the problem [12]. OAC is a flexible, integrative and standardized controller. It consists of four features, namely, extensibility, interoperability, portability and scalability [13]. Major controller suppliers such as GE Fanuc and Siemens have adopted the concept of PC-based open architecture controller. Among the advantages of PC-based control systems in general are: open platform, information handling for improved analysis of manufacturing systems, built-in communication, improved programming and integration into enterprise systems for "e-manufacturing" implementation [14].

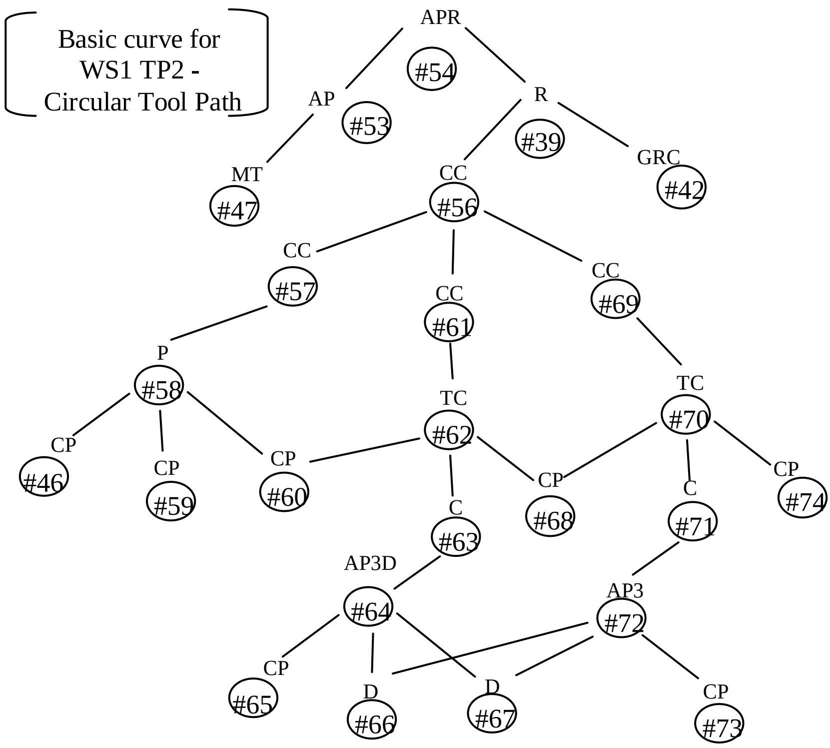
Development of a general parametric curve reference-pulse interpolator which has been successfully implemented under OAPC-NC [15] enables the integration of such system with process planning, CNC and CAD/CAM software.

The Proposed Interface

The main objective of the proposed interface is to extract actual tool path data which can be integrated with a PC-based-open-architecture controller developed by Wan Yusof [15]. A CAD model can be readily exported into STEP AP203 (Configuration Control 3D Design for Mechanical Part) and AP214 (Core Data for Automotive Mechanical Design Processes) using the CAD/CAM system built in translator. Then this file is saved in STEP AP21 physical file format. This physical file is computer interpretable and can be exchanged between various CAD/CAM systems. For this research, the part machining and their CC1 STEP AP238 files are the courtesy of STEP Tools Inc. The interpolator is developed initially for general 2D parametric curve on plane followed by general 3D parametric curve in space.

Tool Path Data Structure

In STEP-AP238 the parameterized path describes parametrically the approach and connect moves of the tool on workpiece so user will know the absolute tool position. It is use when working with cutter contact trajectories for the milling operations. The STEP AP238 file set in this research use cutter location trajectory for its tool path. Thus the extracted data would be in form of the tool center point. Entities are used to define accurately the geometry and topology of a tool path. The information data in STEP AP238 is structured as an inverted tree. A hierarchical approach is proposed for tool path data extraction in this research. Using this approach, the data required is extracted starting from the top element to the bottom element with the same tool path id. The data extracted are the x, y and z coordinates of the tool center point on the tool path, the tool path id in working step (WS) and tool path (TP) number and the connectivity information which showed the line reference number of the data in STEP AP238 file set as shown in Figure 1.



KEY
APR : ACTION_PROPERTY_REPRESENTATION
AP : ACTION_PROPERTY
R : REPRESENTATION
GRC : GEOMETRIC_REPRESENTATION_CONTEXT
MT : MACHINING_TOOLPATH
P : POLYLINE
CP : CARTESIAN_POINT
CC : COMPOSITE_CURVE
CCS : COMPOSITE_CURVE_SEGMENT
C : CIRCLE
TC : TRIMMED_CURVE
AP3D : AXIS2_PLACEMENT_3D
D : DIRECTION

Figure 1: Hierarchy of elements in STEP AP238 for Circular Tool Path

Data Extraction Algorithm

Considering the expandability and adaptability for future works, the interface is modulated following the algorithm shown in following.

Algorithm to extract geometric data from STEP AP238 CC1 part file (Circular Tool Path)

Read STEP file line by line.

While reading strings in file line by line,
  find string matches 'TRIMMED_CURVE'
  store the entire line in cell TC;
  if TC > 0
  then find position of # in TC (i.e. CARTESIAN_POINTs) and record the number of elements found: CP,
    find position of WS in TC (i.e. Working Step) and record the number of elements found: WS,
    find position of TP in TC (i.e. Tool Path) and record the number of elements found: TP,
    DO store WS in 1st column of cell lastnum
    if the fifth character after TP is "," *then* store TP and one number after TP in 2nd column of cell lastnum
    else store two numbers after TP
      or k = 1:sizeCP
      if the 3rd character after # is ",", "=" & ")"
      then store two numbers after # in the 3rd column onwards
      else store three numbers after # in the 3rd column onwards
      end
    end

END (*while* loop ended)

Results and Discussion

For Circular path, the original tool path is derived from the 3D circular-arc equation as given in Equation (1) to Equation (2). However, the entire circular-arc tool path described in the sample data moves on 2D plane.

Therefore, the data for z-axis does not need to be interpolated and the curve tilted angle does not need to be defined.

Start point = (x_s, y_s, z_s), Center point = (x_c, y_c, z_c),

End point = (x_e, y_e, z_e), Rad = Rs

$u_{start} = [(x_s - x_c),(y_s - y_c)]$ $\quad u_{end} = [(x_e - x_c),(y_e - y_c)]$

$$x(u) = -R * \cos(u) \tag{1}$$

$$y(u) = R * \sin(u) \tag{2}$$

Curve boundary $u_{start} \le u \le u_{end}$

The simulation plots are shown in Figure 1 for the 2D circular-arc WS1 TP4 with input data as shown in Table 1. Table 2 presents detailed numerical results.

Table 1: Input Data for 2D Circular-arc Interpolator

WS	TP	Ref Pt.	Start point (SP)	Arc Centre (CP)	Arc End (EP)	Radius, R (mm)
1	4	(#127)	(99.0019, 102.003, 20)	(89.9986, 89.9948, 20)	(104.9966, 90.5564, 20)	15.008

Table 2 Result of 2D Circular-arc for Curve in Figure 1

Iteration (k)	X_k	Y_k	x_k	y_k	$Error_k$
1	10.000	2.000	10.000	2.000	0.000
2	10.000	3.000	9.957	3.000	0.043
3	10.000	4.000	9.876	4.000	0.124
4	9.000	5.000	9.489	5.000	**0.489**
5	9.000	6.000	9.198	6.000	0.198
6	9.000	7.000	8.879	7.000	0.121
7	8.000	8.000	8.211	8.000	0.211
8	8.000	9.000	7.876	9.000	0.124

9	7.000	10.000	7.000	10.000	0.000
10	6.000	11.000	6.000	11.187	0.187
11	5.000	12.000	5.000	12.213	0.213

Figure 2 shows the Interpolator simulation result of 2D circular-arc. The curve is simulated with big step size (BLU=1) so that the curve tracing can be observed. The "exaggerated" plot and data of 3D parametric curve are useful to validate the interpolator method of selecting next interpolated points and to observe the distance of interpolated points from the associated points on original curve. In Table 2, it is observed that the maximum error for 2D circular arc is at point 4 (error = 0.489). This point is below maximum allowable error for 2D curve (0.505BLU).

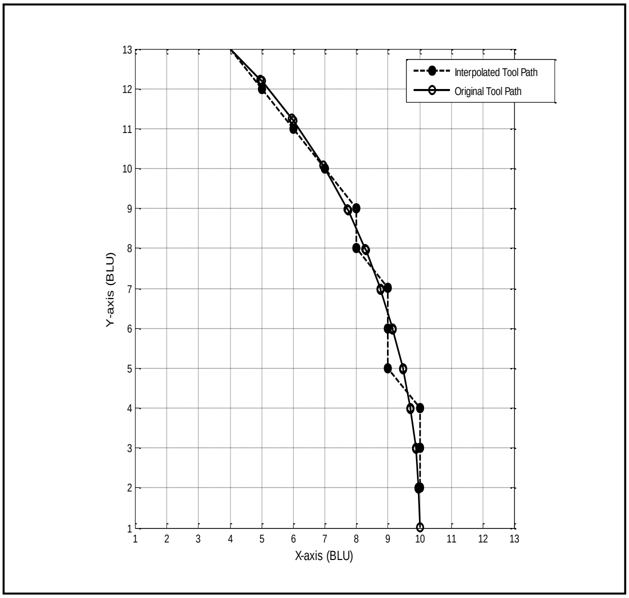


Figure 2: Plot of 'Exaggerated' Interpolator Simulation Result of 2D circular-arc

Conclusions

In relation to the research objectives, the research has successfully developed a tool path data interface utilizing STEP AP238 CC1 neutral file to transfer tool path data from STEP file to OAPC-NC interpolator. Since both the interface and the interpolator are developed and simulated using computers, hence it is portable and adaptive for design and manufacturing activities. Significant contribution from the research work is the interface ability to transfer geometric tool path data from STEP AP238 file to OAPC-NC interpolator.

References

[1] M. P. Bhandarkar, B. Downie, M. Hardwick, and R. Nagi, “Migrating from IGES to STEP: one to one translation of IGES drawing to STEP drafting data,” *Comput. Ind.*, vol. 41, no. 3, pp. 261–277, 2000.

[2] ISO10303-203, “Industrial automation systems and integration - Product data representation and exchange, Part 203. Application Protocol: Configuration controlled 3D designs of mechanical parts and assemblies. 1st Edition, Amendment 1. SIRIM Bhd: ISO10303-203:1994/Amd.1:” 2000.

[3] C. Danjou, J. Le Duigou, and B. Eynard, “Closed-loop Manufacturing, a STEP-NC Process for Data Feedback: A Case Study,” *Procedia CIRP*, vol. 41, pp. 852–857, 2016.

[4] L. Sibul, G. Sziebig, and P. Korondi, “Review of Man-Machine Interaction in Five-axis Machining with STEP-NC,” *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 166–171, 2015.

[5] M. Hardwick *et al.*, “A Roadmap for STEP-NC Enabled Interoperable Manufacturing,” 2013.

[6] X. Zhang, R. Liu, A. Nassehi, and S. T. Newman, “A STEP-compliant process planning system for CNC turning operations,” *Robot. Comput. Integr. Manuf.*, vol. 27, no. 2, pp. 349–356, 2011.

[7] ISO 6983-1, “Numerical Control of Machines-Program Format and Definition of Address Words, Part 1. Data format for positioning, line motion and contouring control systems.” SIRIM Bhd, 1982.

[8] W. Xiao, L. Zheng, J. Huan, and P. Lei, “A complete CAD/CAM/CNC solution for STEP-compliant manufacturing,” *Robot. Comput. Integr. Manuf.*, vol. 31, pp. 1–10, 2015.

[9] ISO 14649, “Industrial automation systems and integration – Product

data representation and exchange. Part 238: Application protocol: Application interpreted model for computerized numerical controllers.," vol. ISO 14649-. 2007.

[10] X. W. Xu, "Realization of STEP-NC enabled machining," *Robot. Comput. Integr. Manuf.*, vol. 22, no. 2, pp. 144–153, 2006.

[11] M. Rauch, R. Laguionie, J.-Y. Hascoet, and S.-H. Suh, "An advanced STEP-NC controller for intelligent machining processes," *Robot. Comput. Integr. Manuf.*, vol. 28, no. 3, pp. 375–384, 2012.

[12] G. Pritschow *et al.*, "Open Controller Architecture – Past, Present and Future," *CIRP Ann. - Manuf. Technol.*, vol. 50, no. 2, pp. 463–470, 2001.

[13] P. K. Wright, "Principles of open-architecture manufacturing," *J. Manuf. Syst.*, vol. 14, no. 3, pp. 187–202, 1995.

[14] X. MA, Z. HAN, Y. WANG, and H. FU, "Development of a PC-based Open Architecture Software-CNC System," *Chinese J. Aeronaut.*, vol. 20, no. 3, pp. 272–281, 2007.

[15] W. A. Wan Yusoff, "Development of an efficient reference-pulse CNC interpolator utilizing general parametric curve," University Science Malaysia, 2003.