The Development of Methodology for Tool Path Generation Using STEP Data Model Based on ISO 6983

Abd Rahman, Z.[a,1], Mohamed, S.B.[a,*], Kasim, M.S.[b,3], Ariffin, M. K. A.[c] and Zulkifli, A.R.[d]

[a]*Faculty of Innovative Design and Technology, UniSZA, Terengganu, Malaysia.*

[b]*Faculty of Manufacturing Engineering, UTeM, Melaka, Malaysia.*

[c]*Faculty of Engineering, UPM, Selangor, Malaysia.*

[d]*Faculty of Engineering and Life Sciences, UNISEL, Selangor, Malaysia*

[1]zammeri@unisza.edu.my, *saifulbahri@unisza.edu.my, [3]shahir@utem.edu.my

Abstract - *The STEP file contains 3D data in a universal data format (ISO 10303-21) structured in boundary representation methodology. It can exchange data models in G-code (ISO 6983) between CAD, CAM, and CNC systems. Currently, G-code generation uses CAM software as an add-on, with high installation costs. This software usually only embraces a specific CAD format. This paper addresses a methodology for designing, implementing, developing, and testing the Integrated Interface System (IIS). In current manufacturing systems requiring intelligent machine controllers, developing IIS for CNC machining provides an alternative to generating G-code without using CAM software. This system can generate G-code based on ISO 6983 and offer tool paths for the machining process using STEP files. The system provides the function to read and extract relevant information from the STEP file that eventually writes a G-code file format. A geometric sample part modeled on a 3D model in a STEP file format. Machining is conducted using the Mach4 controller to mill the sample, and the results were tested using the CNC simulator and the Mach4 simulation. The validation process compares the CAD design features, simulation results, and final machining process products.*

Keywords — *STEP, B-rep, CNC, Geometric Data Extraction, G-code.*

I. INTRODUCTION

A Computer Numerical Control (CNC) machine is a high-tech machine that uses specialized software which serves as the brain to drive and control the machine's axis movement automatically [1]. Its accuracy and high production rate make this machine more preferable than conventional machines in the industry for mass production. Technically, the language behind CNC machining operation is a G-Code based on ISO 6983, which described the motion direction of the tool path corresponding to every axis available for the CNC machine tool. Generally, there two ways of generating a G-Code for CNC machining operation, either manually or automatically generated. The machine operator will interpret the 3D computer-aided design (CAD) design model and write the G-code based on the existing CAD design features for manual operation. But this method is time-consuming and prone to mistakes. CNC machines typically use G-code files for automated processes, but some research findings employ a new approach based on STEP-NC files. These G-code programs generally are generated using add-on CAM tools with CAD tools as inputs [2]. This method, however, has high installation costs, and CAM software normally accepts only specific CAD design formats. CAM must first learn more about the machine to generate G-code for another machine, including model information, comprehensive machine tool specifications, and peripherals such as cutting tools and other parts [3]. The CAM post-processor will handle these components, which will only accept programs in their G-code version, causing incompatibility. Even though the current CNC framework is advanced, data format incompatibility limits the further advancement of CNC-based machining, resulting in data redundancy and potential errors [4].

However, one method for automatically generating G-code is to scan the STEP file. The STEP (ISO 10303-21) was developed by the International Standards Organization (ISO) as a global standard in extending STEP to counter proprietary data limitations between CNC machines by providing a new data interface among CAD / CAM and CNC systems [5]. STEP files are an international standard that contains CAD model design features and available tools for exchanging data models in a CAD format between standard CAD systems and applications [6]. STEP is a text-based universal data structure that can be opened, edited, and saved as an ASCII text file [7]. It contains geometric information from three-dimensional CAD models in B-rep format. These geometric data can generate G-code for CNC machining if intelligently extracted, filtered, refined, manipulated, and decoded. Data in the STEP file format is

extracted repeatedly in a specific order [8]. Understanding the hierarchy of geometric data is required for interpreting a STEP file, and each keyword represents a geometric object [9]. In general, not all data in the STEP file is needed for the generation of tool paths. Only those related to the coordinates and only data related to machining features are required. Many types of research on automatic feature recognition using STEP files have been conducted. Still, before this research, no researchers were investigating practical systems involving the introduction of full profile features in STEP files and translating them into G-code. Even the conversion of a format to G-code, particularly STEP to G-code, is still lacking. With these constraints, developing a new methodology for tool path generation using STEP files based on ISO 6983 is critical. This integrated interface system (IIS) was designed to bridge that gap.

II. STEP-DATA MODEL

STEP is the next-generation data model for CAD, CAM, and CNC. It is a neutral CAD file that contains 3D data in the ISO 10303 universal data format. It is made up of computer-aided design (CAD) geometric data in boundary representation (B-rep) [10]. STEP is a high-level programming language whose data contains solid models and features produced with a CAD system [2]. STEP is written in an ASCII structure in EXPRESS modeling language for easy reading and encryption. It provides bidirectional data flow without redundancy or loss of information between the CAD/CAM and the CNC. The STEP file structure consists of linguistic and unambiguous context-free grammar to aid software parsing. The STEP language is represented in Wirth Syntax notation, and the file information is in an accessible format that does not rely on the column [8]. The ISO 10303-21 STEP file format extension is a STEP or STP file. The file is neutral open-source that can be accessed with a regular text editor, such as Sublime Text, Notepad, Textfile, Word, etc. The STEP file is started with the ISO-10303-21 keyword, ended with the END-ISO-10303-21 keyword, and sections are similarly described as keywords [3], [6]. Much very complex data needs to be analyzed between these two keywords and categorized into three types: Descriptive, Geometric, and Topological. These data types are arranged in two main sections: the "HEADER" section and the DATA section. Section HEADER describes the general STEP information [11]. It contains descriptive data are providing information about the STEP Application Protocols (AP) type in FILE_DESCRIPTION.

Instead, it includes the file name in FILE_NAME, the date and time of creation, the STEP translator version, the CAD modelling software, and the type of file schema used in FILE_SCHEMA [12]. Meanwhile, the DATA section is rich in information and contains a 3D CAD model's geometric data in the visual of B-rep. An example of the STEP AP203 file HEADER and DATA section. The STEP

file consists only of solid modeling features containing geometric and topological data [4]. It consists of several definitions of geometrical entities and certain topological elements, such as faces, loops, and borders. It can be shown in the example of a rectangular sample block, as shown in Fig. 1. The solid surface of a rectangular block is a combination of CLOSED_SHELL. This CLOSED_SHELL consists of specific ADVANCED_FACE, where every ADVANCED_FACE is bound by several edges known as FACE_BOUND or FACE_OUTER_BOUND.
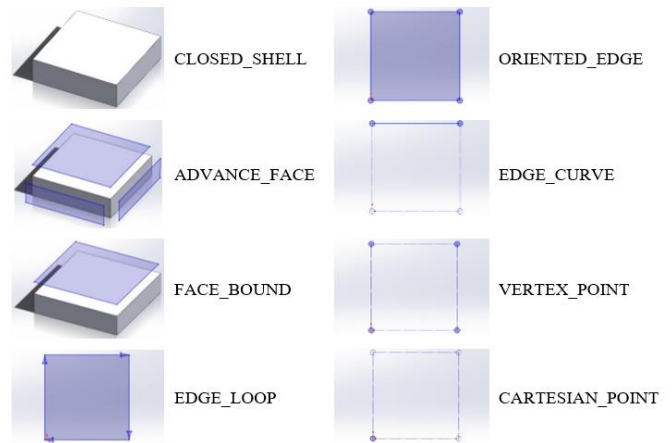
**Fig. 1 Sample block entities and elements for the hierarchical tree of the STEP file**

Simultaneously, each FACE BOUND will be formed with an edge as a loop called EDGE_LOOP. Each of these EDGE_LOOPs is then formed by ORIENTED_EDGE, which is termed as EDGE_CURVE. Each EDGE_CURVE present is formed from a combination of several VERTEX_POINT corners. Each VERTEX_POINT is formed from CARTESIAN_POINT, consisting of three axes: x, y, and z. The 3D CAD solid model's core data are geometric entities and topological elements nestled in the data section [13]. In AP203 files, data entities and elements are designed to control the product model's configuration. This paper does not provide a detailed description of the definitions of geometrical entities and topological elements. In ISO/TC/184/SC4 standards, the respective reports and contributors are available. STEP is relatively new in CNC machining and aimed to unify the flow of geometric information from the 3D CAD model to the NC controller. Most CAD systems can define and use STEP file data as in the B-rep model. These data are helpful when extracting, filtering, manipulating, and decoding G-codes. However, not all STEP file data able to use for CNC G-code programming. The post-processor is no longer required when using this STEP-data input file because the G-code generated and the machining process is sufficient [2]. While STEP is still under research and development, it is perfectly functional to represent a model's geometric data. The STEP

standard emerges as a highly applicable method to further expand CAx chains' compatibility to more advanced CNC levels. As the file data model structure, STEP is also evolved worldwide, and it will become more flexible and more geometrical.

III. PROPOSED METHODOLOGY

The methodology adopted in the paper is intended to integrate CAD and CNC to avoid CAM post processors. The

profile's path pattern can be calculated and recognized with the final information available, whether straight or curved. The IIS is a program that was developed to use input from the STEP file and was designed as a user interface system that can convert data from the STEP file to a G-code program. The system is web-based, designed with PHP programming language using the web browser control, and accessed on a localhost webserver. It requires the user to provide additional inputs such as spindle speed, feed rate,
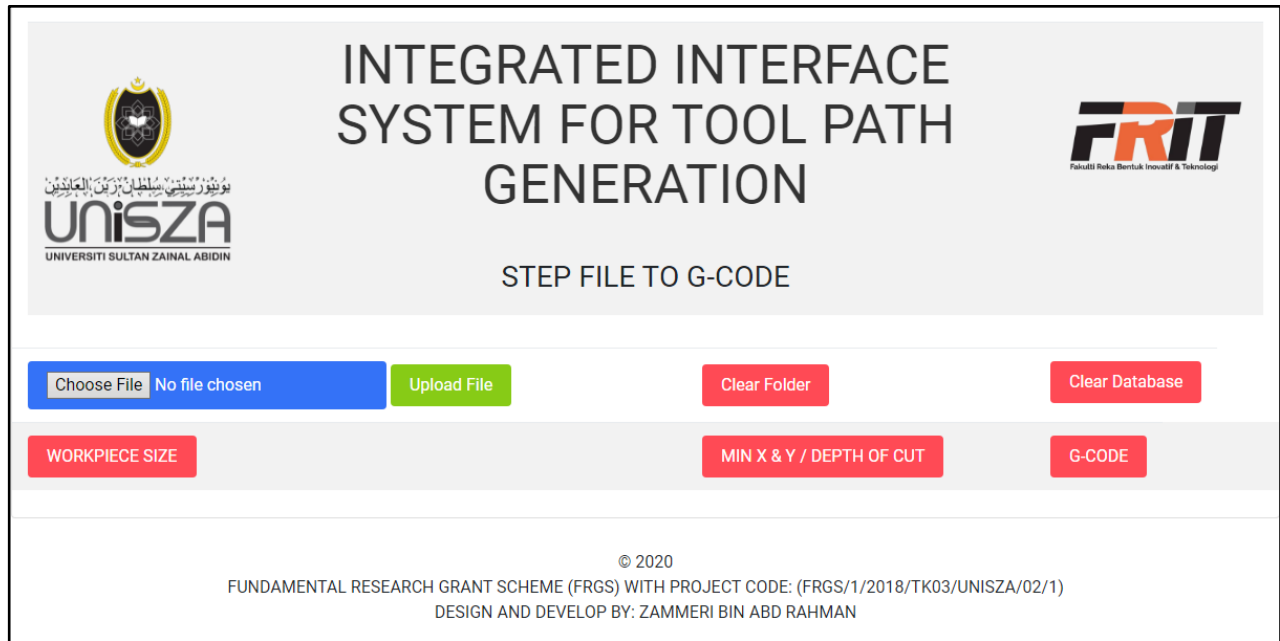
Fig. 2 Integrated Interface System (IIS)

sample model is designed with 3D CAD modeling software to be used as the main product feature. A simple model has been designed for a fully defined solid design and is stored in a STEP file. It is necessary to use the STEP file because it is compatible with several known CAD systems. This research uses the Part-21 STEP AP203 file to exploit the cartesian coordinates to generate the G-code [2]. STEP data is analyzed, and Geometric Data Extraction (GDE) is performed to classify existing geometric data. The GDE algorithm is developed based on a rule-based approach that simplifies and facilitates data representation and removal [14]. In this process, it is essential to create a STEP file structure in determining the tree pattern of the STEP file hierarchical tree. The process interprets the B-rep information obtained by translating the geometric data into a B-rep data structure in the next phase. This method was created to build an algorithm to extract geometrical information data from a STEP file. The algorithm specifies the kind and orientation of the face by employing a B-Rep database with specifics on geometrical and topological information, and it ends with cartesian points [13], [15]. The

cutting tool, and G/M code as raw data needed to generate G-code, as presented in Fig. 2.

This IIS system architecture consists of six data modules: read/write/rewrite, extraction, elimination, processing, integration, and transformation. The read-write/write module is the program framework that reads all solid features and saves them as the required file when saving the 3D model in a STEP file format. This method interprets the input data string until the process completes and preps it in the next loop. The writing process is when the solid model's information is represented in a text format containing all the given model's design functions. Later, the rewrite occurs when the cartesian coordinate is extracted from the STEP file, and the output is when the cartesian coordinate is used to produce the G-code [16]. The data extraction module will analyze each file line to identify and extract all geometric data in the STEP file, then write and record it in the database. The data elimination module screens and eliminates duplication, redundant, and recurrent data from the database that is not required to generate a tool path.

Furthermore, all information is processed by data processing modules according to the predefined set of algorithms. Processing algorithms have three submodule operations for conducting operations of extraction, elimination, processing, and integration. The data integration module integrates the user's required specific parameters to provide additional inputs for the tool path's generation, including spindle speed, feed rate, cutting tool, and G/M code [17]. A complete G-code generation process produces a sequence of axis motion through a transformation module. Tool functions are made from the processed integration data in the Tool Path Generation (TPG) setup [17]. The IIS conceptual architecture system will automatically generate the G-code based on ISO 6983 for machining tool path movement referring to 3D solid model CAD design in the process control layer. The resulting G-code is tested and validated with a tool path simulation process in Artsoft Mach4 software as a motion controller software. This software is also known as a PC-based open architecture controller software, and all these processes are performed in the coordination layer. Finally, the machining process is performed by milling the sample parts that allowing point-to-point operation profile cutting processes using a 3-Axis CNC milling machine. This process is completed at the functional layer, and methodology is illustrated as a framework, as shown in Fig. 3.

IV. DEVELOPING THE PARSER OF STEP FILE

The STEP file contains three primary geometric data components: B-rep data, geometric surface data, and geometric curves. The data's B-rep includes shells, faces, edges, loops, and vertex. Simultaneously, geometric surface data contains planes and cylinder entities [18]. The line and circles entities are geometric components and parts of the geometry curve data. The STEP file data represented by an entity line does not display in a logical order. The file is nearly unreadable since it involves the non-sequential line-to-line transfer [19]. It is not only illegible to the programmer but also for the geometric recognition of features. Therefore, essential to create a parser to rearrange the data in an intelligible manner. The features recognition system and the shape of the STEP files typically began with the CLOSED_SHELL string. It ended with the CARTESIAN_POINT of the geometric coordinate string [20]. As shown in the hierarchical structure tree in Fig. 4, other strings include ADVANCED-FACE, FACE-BOUND, and so on. In this research, the STEP file is imported into the STEP AP203 parser. The program scanned all its lines to decide how many CLOSED SHELL and faces contains in the geometric string. Instead, each face is shown with its geometrical and topological detail before moving to the next face. When an ADVANCED_FACE is marked, the

STEP AP203 / AP214 (ISO 10303-21)

CAD Database 3D Solid Model

User / Programmer

GEOMETRIC DATA EXTRACTION (GDE)

Identify STEP File Data Structure
Define STEP File Hierarchical Tree
Interpretation of B-rep
Recognize Profile Path
Circular — Linear

ALGORITHM

Rule Based Approach

INTEGRATED INTERFACE SYSTEM (IIS)

Read/Write/Rewrite Module
Extraction Module
Elimination Module
Processing Module
Integration Module
Transformation Module

DATA

User Input: Spindle Speed/Feed Rate/ Cutting Tool/G & M Code

G-Code (ISO 6983)

Mach4 CNC Motion Controller

3-Axis CNC Milling Machine

Coordination Layer — — **Process Control Layer** — — **Functional Layer** — —

Fig. 4 Geometric string hierarchy of the STEP file

FACE_BOUND will appear in the following line, which could be either FACE_BOUND or FACE_OUTER_BOUND. In the ADVANCED_FACE string, an entity PLANE will occur if a couple of FACE_BOUND and FACE_OUTER_BOUND entities are present in a closed parenthesis. Next, the parser indicates an EDGE_LOOP object with an edge indicator to form the face [19]. All edges are provided up to the lowest level of the CARTESIAN_POINT entity with relevant information. The parser will be shown as the previous one when the face has another bond. When the face is in contact with another bond, the parser will display it in the same manner. Otherwise, the face will indirectly state all the specifics before moving to the next face. The PHP programming language is used to create the parser, which aims to recognize features in the quest for unique strings and entities in the STEP file. The creation algorithm was developed based on a rule-based technique, making data representation and extraction simple and easy. However, before proceeding to the extraction section, more manipulation, editing, and filtering are required. This process is accomplished through the use of the algorithms discussed and implemented in this research.

III. INTERPRETATION OF B-REP

The B-Rep model, usually based on a surface representation. The description of shapes by adding geometric constraints is a technique that applies to B-rep [21]. The B-rep model consists of two types of geometric information to define spatial position and curvatures.

Fig. 5 3D solid wireframe model represented according to a B-Rep

A B-rep model involves geometric entities for specific

components, including surfaces, curves, and points [23]. Together with topological entities include the faces, edges, and vertices or the relations between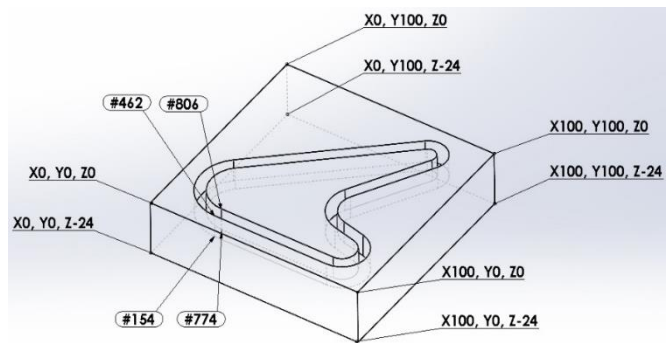 them. The face is the part of the bounded surface, while the edge is a piece of bounded indentation, and the vertex is located at one point. There is a shell in which a group of faces is connected [24]. The loop is a circuit with edges that bounding a face. The loop edge link, also known as the winged edge link or half-edge, is used to construct edge circuits. The GDE algorithm can identify the sample's pattern in the analysis by defining the block size by finding each corner and coordinate point. Feature recognition is used to evaluate specific patterns and shapes by identifying all STEP files' geometric data information. As shown in Fig. 5, this concludes that block size is 100mm (width) x 100mm (length) x 24mm (height). This illustration show has the 3D solid features and in wireframe models. Data extraction is also used to describe the profile path pattern by extracting and identifying the geometry data or the elements and entities present. That allows the all present EDGE_LOOP (#154, #462, #774, #806) to be determined that complements each other in forming a close profile.

VI. GEOMETRIC DATA EXTRACTION ALGORITHM

Following a thorough review of the STEP file, efforts are made to design an integrated interface system by developing geometric data extraction algorithms based on the B-Rep derived from the STEP AP203 file. The developed algorithm will use the B-Rep database to determine each face type and orientation [13], [15]. Include both topological and geometric object information and conclude by compiling the information present in each cartesian point. Identifying the algorithm's pattern and design in the STEP file is the most important part and requires very detailed evaluation [25]. Several methods are used from the beginning of the study, such as conversion, recognition, extraction, integration, elimination, data processing, read/write/rewrite, generation, comparison, and validation. All these methods can help and accelerate the data parsing process for G-code generation. A few samples were designed as validation processes in this study, and the Sample model was used to complete the methodological approach and apply all the stated methods. For analysis, all these cartesian points are collected and extracted. G-code is
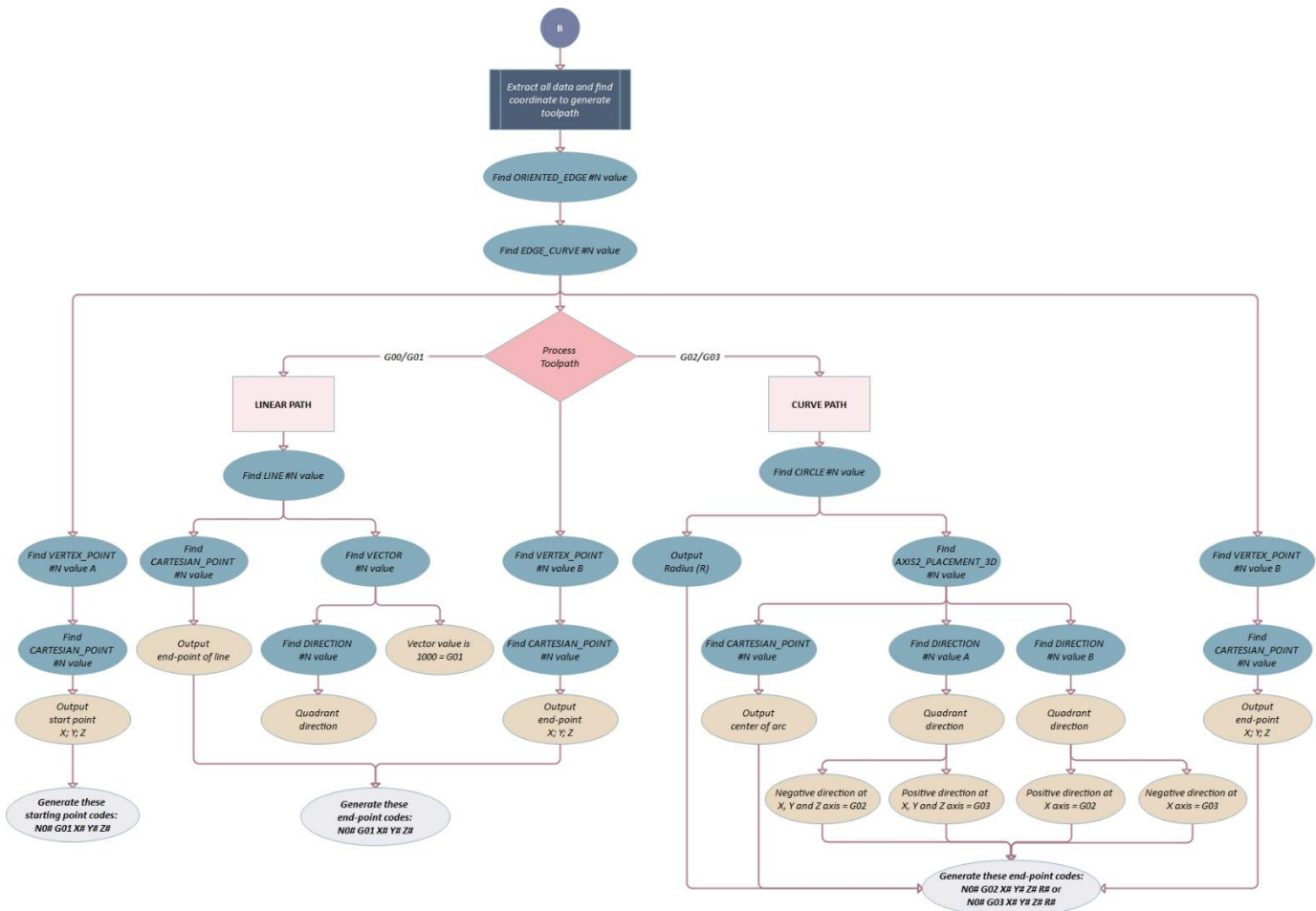
Fig. 6 The framework of the profile path

generated using coordinates for each axis, and all coordinates are combined to create machining tool paths. The coordinate point transition is a cutting tool's movement, which generates a tool path based on the geometric feature information [26], [27]. The general PHP language program is used and written to extract and classify geometric information from STEP files based on specific framework models. These frameworks are identified by clusters divided into four sections: workpiece size, min x & y, depth of cut, an edge loop path, and profile path. This algorithm was developed and compiled into a complex integrated interface system that can generate G-code automatically. The IIS has been designed by identifying several factors, including manually reading the operator's input and the header or footer's input code. The specified inputs include spindle speed, feed rate, cutting tools, G code, and M code.

VII. DATA EXTRACTION OF THE PROFILE PATH

The suggested framework in Fig. 6 can be the method to develop automatic G-code generation. Several ways were adopted by reading geometric information and extracting the data step-by-step. This framework may also describe two paths to remove the profile path pattern for a curved and the profile path for a straight line. According to the programmed analysis, this method generates output rows and is ordered by set "N". The G-code, either G00, G01, G02, or G03, ends with CARTESIAN_POINT, which defines each axis x and y. Simultaneously, the z-axis represents the reading of the cutting depth. For G-codes, G02 and G03 represent tool path movement along with radius. The task is interpreted by defining the start and endpoint, followed by the center of the

#154 = EDGE_LOOP (#595, #649, #379, #904, #82, #590, #363, #654, #375, #851)

Edge Loop Path

Depth of Cut = Z-10

Tool Path Point

Tool Path Pattern

Fig. 7 Tool path motion in EDGE_LOOP extraction

arc. This radius is written as R in the G-code line display.

By determining the EDGE_LOOP entity, all #N entities and geometric data are extracted for identifying the CARTESIAN_POINT coordinates to generate a tool path. This approach is conducted in stages by defining all present entities, as presented in Fig. 7. Once EDGE_LOOP was identified, and the extracted data will be associated with ORIENTED_EDGE. After that, EDGE_CURVE will be placed to determine the entities that have context between them. Upon arrival at this step, three entities are closely related, such as VERTEX_POINT, LINE, and CIRCLE. These three entities can decide the G-code for the tool path's motion, whether moving counter-clockwise, clockwise or linear. Once the tool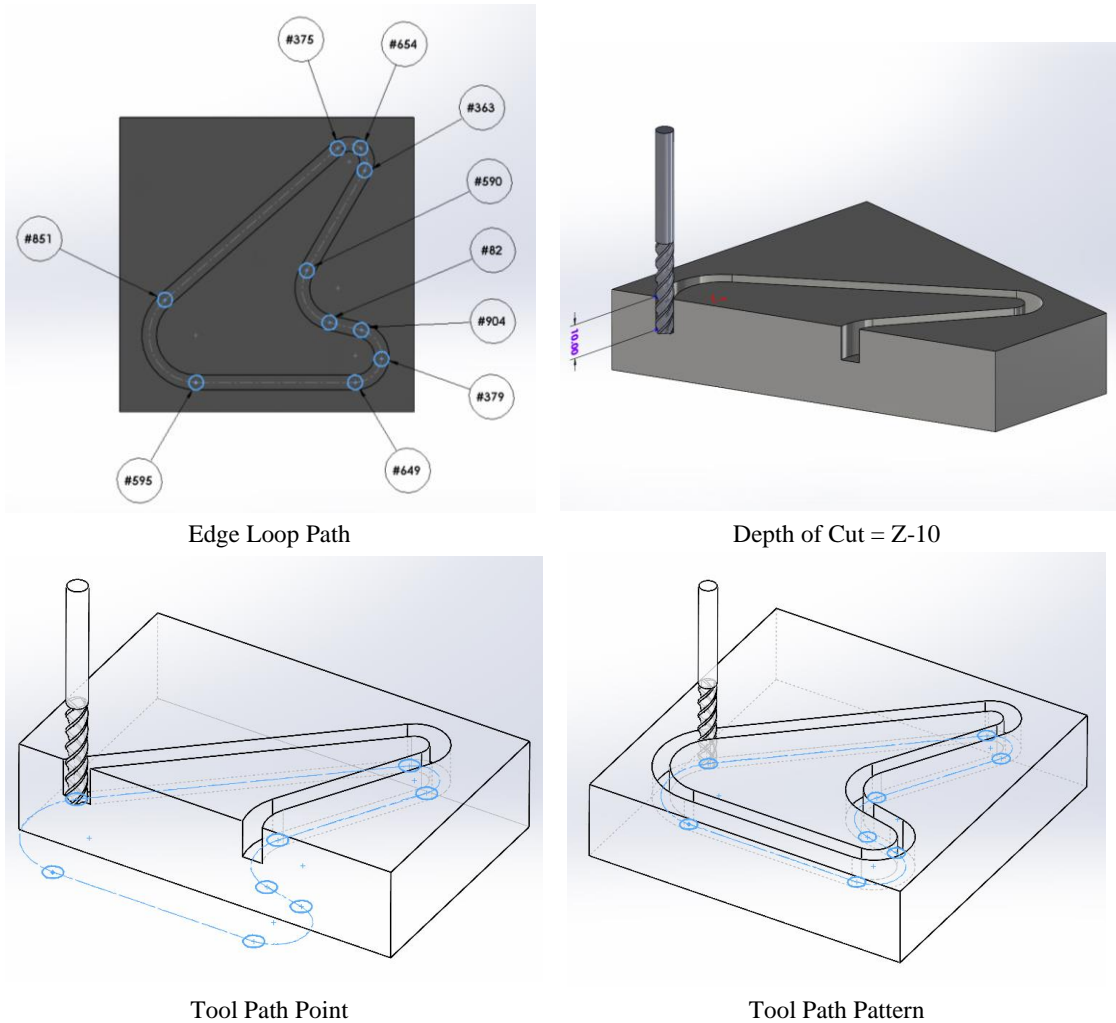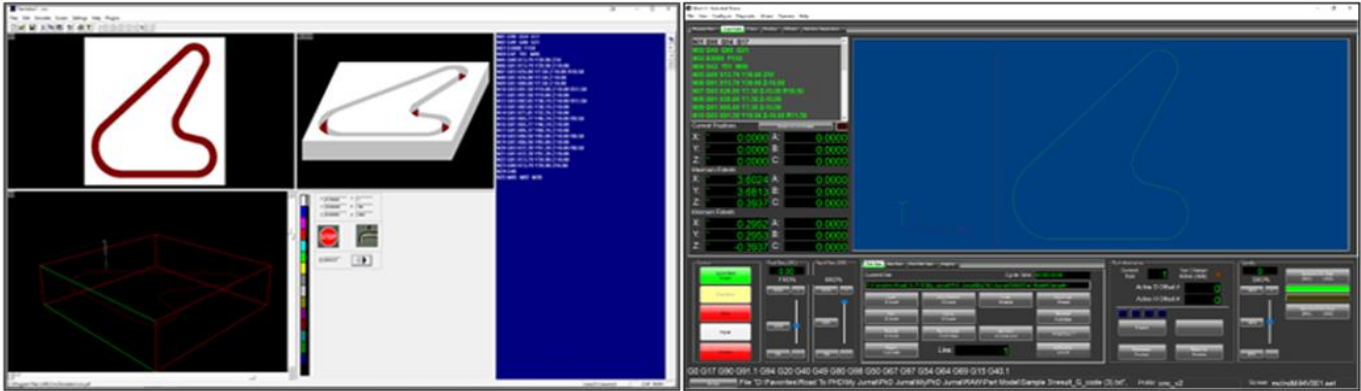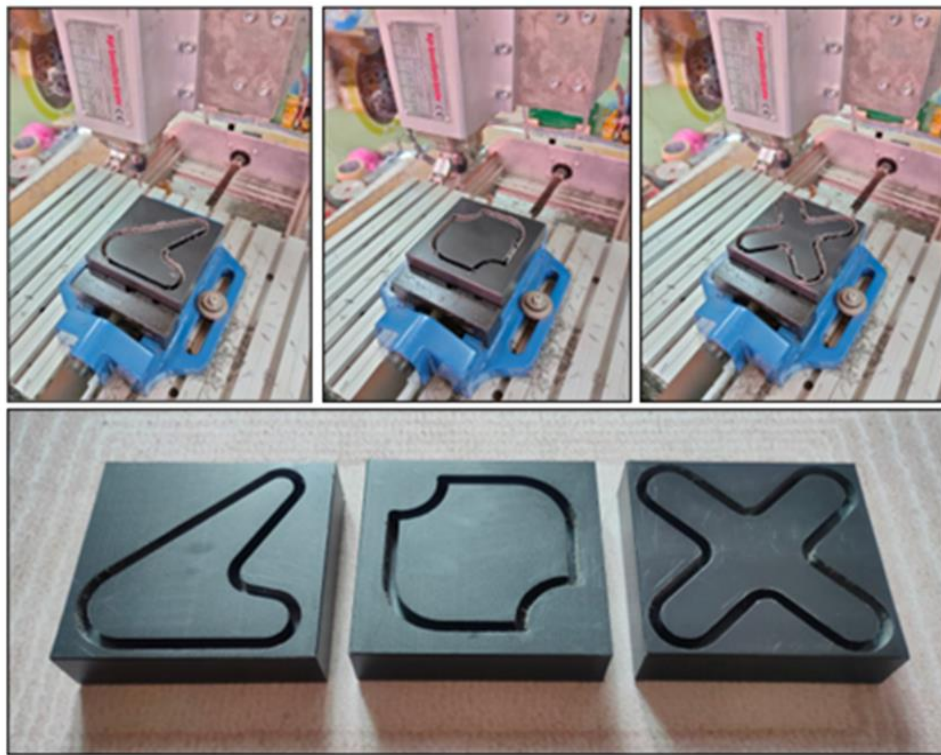 path's movement point can be established, this entity will determine the next reading of four distinct entities with their functions: CARTESIAN_POINT; VECTOR; DIRECTION; and AXIS2_PLACEMENT_3D. The G-code determinant is essential to the tool path's movement, whether it moves counter-clockwise, clockwise, or linear. Each cutting tool

reading indicates G03 counter-clockwise motion. In that case, the first DIRECTION will display a positive value reading on the x, y, and z-axis coordinates. Otherwise, the second DIRECTION will show a negative value reading on the x-axis coordinates. For tool path direction readings that indicate G02 clockwise motion, the first DIRECTION will display a positive value reading on the x-axis coordinates. Instead, the second DIRECTION will show a negative value reading on the x, y, and z axes coordinates. However, for motion readings G03 and G02, the CIRCLE entity's presence determines the radius of circles at the second reading in parentheses. The program will define three different LINE, VECTOR, and DIRECTION entities to assess the tool path moves in a G01 linear direction. The VECTOR will display a reading of 1000.00 in parentheses on the second value. In contrast, the DIRECTION entity will show a positive reading on the x and y-axis coordinates, as illustrated in Fig. 8. At the end of the reading, the identified data will be programmed according to the international standard arrangement ISO 6983, and the output will be displayed as (G_{XX} X, Y, Z, R_X).

Data Extraction for G03

#209 = VERTEX_POINT ('NONE', #605) ;
#632 = VERTEX_POINT ('NONE', #688) ;
#502 = *CIRCLE ('NONE', #92, 18.49999999999999645) ; Curve Path with Radius*
##605 = CARTESIAN_POINT ('NONE', (13.78916790503032708, 39.89768252438004481, -10.0000000000000000)) ; *First Line*
#688 = CARTESIAN_POINT ('NONE', (26.00000000000000355, 7.50000000000001776, -10.0000000000000000)) ; *Second Line*
#92 = AXIS2_PLACEMENT_3D ('NONE', #349, #673, #701) ;
#349 = CARTESIAN_POINT ('NONE', (26.00000000000000000, 26.00000000000000000, -10.0000000000000000)) ; *Center of Arc*
#673 = DIRECTION ('NONE', (0.000000000000000000, 0.000000000000000000, 1.00000000000000000)) ; *+VE Direction at X Y & Z axis = G03*
#701 = DIRECTION ('NONE', (-1.00000000000000000, 0.000000000000000000, 0.000000000000000000)) ; *-VE Direction at X axis = G03*

Data Extraction for G02

#550 = VERTEX_POINT ('NONE', #642) ;
#424 = VERTEX_POINT ('NONE', #434) ;
#476 = *CIRCLE ('NONE', #680, 9.500000000000008882) ; Curve Path with Radius*
#642 = CARTESIAN_POINT ('NONE', (71.81218933352793954, 32.75535373918118864, -10.0000000000000000)) ; *First Line*
#434 = CARTESIAN_POINT ('NONE', (65.76841742071792396, 46.74247279807276101, -10.0000000000000000)) ; *Second Line*
#680 = AXIS2_PLACEMENT_3D ('NONE', #622, #836, #764) ;
#622 = CARTESIAN_POINT ('NONE', (74.00000000000001421, 42.00000000000000000, -10.0000000000000000)) ; *Center of Arc*
#836 = DIRECTION ('NONE', (-0.000000000000000000, -0.000000000000000000, -1.00000000000000000)) ; *-VE Direction at X Y & Z axis = G02*
#764 = DIRECTION ('NONE', (1.00000000000000000, 0.000000000000000000, 0.000000000000000000)) ; *+VE Direction at X axis = G02*

Data Extraction for G01

#424 = VERTEX_POINT ('NONE', #434) ;
#733 = VERTEX_POINT ('NONE', #528) ;
#223 = *LINE ('NONE', #8, #669) ; Straight line Path*
#434 = CARTESIAN_POINT ('NONE', (65.76841742071792396, 46.74247279807276101, -10.0000000000000000)) ; *First Line*
#528 = CARTESIAN_POINT ('NONE', (85.36510020251554920, 80.75673486488229003, -10.0000000000000000)) ; *Second Line*
#8 = CARTESIAN_POINT ('NONE', (85.36510020251554920, 80.75673486488229003, -10.0000000000000000)) *End Point of Line*
#669 = VECTOR ('NONE', #143, 1000.000000000000227) ; *Linear Path = G01*
#143 = DIRECTION ('NONE', (0.4992076629550265543, 0.8664823767665332799, -0.000000000000000000)) ; *+VE Direction at X & Y axis = G01*

Fig. 8 Data extraction for tool path motion

path entity is read with G03 in the counter-clockwise direction, G02 in the clockwise direction, and G01 in the linear direction. Several methods can determine the tool path movement according to the DIRECTION, VECTOR, CIRCLE, and LINE readings present in each entity from the geometric data extracted. Suppose the tool path direction

VII. MACHINING VALIDATION AND TESTING

As illustrated in Fig. 9, the tool path is simulated using a CNC simulator before being tested with a Mach 4 Industrial. The G-code verification tests were generated from IIS and

Fig. 9 G-code viewer and machining tool path simulation

Fig. 10 Machining process and testing

tested using Mach4 according to the required setting parameters. Mach 4 is preferred as a PC-based CNC controller and as an open-source CNC controller. It can operate without relying on specific machine manufacturers and proprietary software vendors. Before applying in the milling process, this application generates the tool path by uploading the 3D CAD design in the wireframe model. The simulation results' features are compared to the CAD design features and the machining process's end product. The 3-axis CNC milling machine is used for physical machining tests earlier applied in G-code validation simulation. The Ultra High Molecular Weight Polyethylene (UHMW-PE) sample with a 100mm x 100mm x 24mm dimension size is used as a machining workpiece. A 3-axis CNC milling machine configured with Mach4 Industrial, which is an open-source CNC controller. For the machining test, the spindle speed is set to S3000, and feed rates are set to F500. Fig. 10 depicts three machining samples, each with a trial of a different shape. The x, y, and z coordinates of the automatically generated G code correlated with the STEP file's cartesian coordinates. A machining test is performed to determine the tool path accuracy, whether in the curve or straight cut in the workpiece. This combination of cutting patterns creates a profile path that proves this IIS ability to generated G-code in various forms.

VI. CONCLUSIONS

According to the findings of this research, the IIS thrived generates a tool path using a STEP-data model based on the ISO G-code standard. Furthermore, it no longer requires CAM software and can avoid using a post-processor because machine-specific details are no longer needed. To evaluate 3D CAD models, it is no longer necessary to use proprietary or closed source software. It also eliminates the need for additional CAM tools and their high setup costs. Moreover, the method enables beginners to learn and train to use CNC machines, thereby growing the skilled workforce. This geometric coding research knowledge will aid in the comprehension of geometric data entities in STEP files. For enhancing future research, it is expected that new research can be explored and transformed the simple profile path into various and more complex forms. With this, the algorithm generated for the tool strip from the STEP file will be more complicated. The diversification of methods can be developed further for geometric data extraction in features recognition and topological data in STEP files from this interface system. Exploring this developed system's capabilities would be a more significant and fascinating challenge.

ACKNOWLEDGMENT

REFERENCES

[1] M. Hou and T. N. Faddis, Automatic tool path generation of a feature-based CAD/CAPP/CAM integrated system, International Journal of Computer Integrated Manufacturing. 19(4) (2006) 350-358.

[2] Mohamed, S. B., Minhat, M., Sulaiman, T. M. S. T., Kasim, M. S. M., Mohamed, A. S., Mohamed, A. R. and Ismail, J., Integrated interface system for tool path generation of STEP file, AIP Conference Proceedings. 2129 (2019).

[3] Z. Al-Bayaa, M.Minhat, V.Vyatkin, X.Xu and S.Wong, A novel open CNC architecture based on STEP, Robotics and Computer-Integrated Manufacturing. (2008).

[4] T. M.S. Tg Sulaiman, S. B. Mohamed, M. Minhat, A. S. Mohamed and A. R. Mohamed, Integrated Interface Development Environment using STEP Universal Data Structure, International Journal of Engineering & Technology. 7 (2.15) (2018) 27.

[5] Emad S. Abouel Nasr and Ali K. Kamrani, A New Methodology for Extracting Manufacturing Features From CAD System, Computers & Industrial Engineering. 51(3) (2006) 389-415.

[6] X. Huang, G. Zhou and R. Liang, A Web 3D Viewer Based on STEP, 16th International Conference. (2007).

[7] N. Ismail, C. F. Tan, S. V. Wong, M. R. Osman and S. Sulaiman, Rule-based feature extraction and recognition from STEP file, 2002 Student Conference on Research and Development: Globalizing Research and Development in Electrical and Electronics Engineering, SCOReD 2002 - Proceedings. (2002) 90-93.

[8] M. Liang, S. Ahamed, and B. van den Berg, A STEP based tool path generation system for rough machining of planar surfaces, Computers in Industry. 32(2) (1996) 219-231.

[9] P. L. Wenlei Xiao, Lianyu Zheng, and Ji Huan, A complete CAD/CAM/CNC solution for STEP-compliant manufacturing, Robotics and Computer-Integrated Manufacturing. (2014).

[10] A. A. Khan, E. A. Nasr, and A. Al-Ahmari, An independent CAPP system for prismatic parts, International Journal of Rapid Manufacturing (IJRAPIDM). 5 (2015).

[11] C. Danjou, J. Le Duigou, and B. Eynard, Closed-loop manufacturing process based on STEP-NC, International Journal on Interactive Design and Manufacturing (IJIDeM). 11 (2017) 233-245.

[12] V. N. Malleswari, P. Valli M, and M. M. M. Sarcar, Automatic Recognition of Machining Features using STEP Files, International Journal of Engineering Research & Technology (IJERT). 2(3) (2013).

[13] D. Sreeramulu and C. S. P. Rao, A new methodology for recognizing features in rotational parts using STEP data exchange standard, International Journal of Engineering, Science and Technology. 3(6) (2012) 102-115.

[14] B. Babic, N. Nesic, and Z. Miljkovic, A review of automated feature recognition with rule-based pattern recognition, Computers in Industry. 59(4) (2008) 321-337.

[15] D. Sreeramulu and C.S.P. Rao, A New Methodology for Recognition of Milling Features from STEP File, The International Journal of Applied Management and Technology. 6(3) (2008).

[16] S. B. Mohamed, A. Jameel, and M. Minhat, A Review on Intelligence STEP-NC Data Model and Function Blocks CNC Machining Protocol, Advanced Materials Research. 845 (2013) 779-785.

[17] S. B. Mohamed, D. J. Petty, D. K. Harrison, and R. Rigby, A cell management system to support robotic assembly, International Journal of Advanced Manufacturing Technology. 18(8) (2001) 598-604.

[18] S Ma, Y Maréchal and J.-L Coulomb, Methodology for An Implementation of The STEP Standard: A Java Prototype, Advances in Engineering Software. 32(1) (2001) 15-19.

[19] M. Al-wswasi and A. Ivanov, A novel and smart interactive feature recognition system for rotational parts using a STEP file, International Journal of Advanced Manufacturing Technology. 104(1-4) (2019) 261-284.

[20] N. Ismail, S. V. Wong, S. Sulaiman, and M. R. Osman, Development of Hole Recognition System from Step File, ASEAN Journal on Science and Technology for Development. 22(3) (2007) 285-295.

[21] J. Wang, Research on Shape Feature Recognition of B-Rep Model Based on Wavelet Transform, Mathematical Problems in Engineering. (2018) 8.

[22] M. A. R. Md Sharizam, S. B. Mohamed, T. M. S. T. Sulaiman, Z. Abd Rahman, and R. Awang, Development of the PC-based Integrated Interface System of STEP file for CNC Machining Application: Circular Features, International Journal of Integrated Engineering. 12(7) (2020) 195-203.

[23] D. Sreeramulu, S. P., and S. K. Adapa, A New Methodology to Extract Machining Features From Prismatic Parts, in International Conference on Advances in Engineering and Technology (ICAET'2014). (2014).

[24] P. Sateesh and P. V Mahesh, A Methodology for Feature Extraction and Recognition For CAD/CAM Integration Using Step File, Mechanical Engineering IJRIME. 469(3) (2017) 319–323.

[25] C. F. Tan, N. Ismail, S. V. Wong, S. Sulaiman, and M. R. Osman, Development of Step Hole Recognition System for Computer Aided Process Planning, Jurnal Teknologi. 43(1) (2005) 73–86.

[26] A. Anbalagan, Feature recognition & tool path generation for 5 axis STEP-NC machining of free form / irregular contoured surfaces, Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium. 21(1) (2010) 1235–1236.

[27] C. F. Tan, V. K. Kher, and N. Ismail, Design of a feature recognition system for CAD/CAM integration, World Applied Sciences Journal. 21(8) (2013) 1162–1166.