© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

# Integrated Interface System (I$^2$S) of STEP Data Models for the CNC Milling Tool Path Generation

**Zammeri Abd Rahman[1], Saiful Bahri Mohamed[1*], Mohammad Minhat[2], Alias Mohd[3], Zulkifli Abd Rahman[3]**

[1]Faculty of Innovative Design and Technology,
University Sultan Zainal Abidin (UniSZA), Terengganu, 21300, MALAYSIA

[2]Faculty of Manufacturing Engineering,
University Teknikal Malaysia Melaka (UTeM), Melaka,76100, MALAYSIA

[3]Faculty of Engineering Technology,
University College TATI (UC TATI), Terengganu, 24000, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2022.14.04.012
Received 27 November 2020; Accepted 6 December 2020; Available online 20 June 2022

**Abstract:** The STEP (STandard for the Exchange of Product model data) is a worldwide standard for exchanging products between diverse computer systems and industrial applications. STEP files (ISO 10303) contain 3D data in the universal data format with a boundary representation (B-rep) that allows the exchange of data models between CAD, CAM, and CNC systems. Traditionally, G-code generation relies on CAM software as an add-on, which incurs significant installation costs. The purpose of this paper is to describe the development of an Integrated Interface System (I$^2$S) for Tool Path Generation (TPG) for the CNC milling operations utilizing neutral STEP files. The algorithm was interpreted in Hypertext Preprocessor (PHP) programming and produced a computer interface system that is capable to convert STEP file geometric data to G-code format. The geometric features of the sample block were modeled using a 3D CAD model. The result was validated using machining simulation, and the machining is performed on a three-axis CNC milling machine with a PC-based controller. In modern manufacturing systems based on Open Architecture Controller (OAC), the development of the I$^2$S provides an alternative to generating machining tool paths and G-code by intercepting CAM software.

**Keywords:** STEP, Integrated Interface System, CNC Milling, Tool Path, G-code

## 1. Introduction

The STandard for the Exchange of Product data model (STEP) serves as the next-generation data model for Computer-Aided Design (CAD), Computer-Aided Manufacturing (CAM), and Computer Numerical Control (CNC) systems in the form of a neutral CAD file [1]. STEP is a high-level programming language that incorporates solid models generated by 3D CAD software. The STEP file contains three-dimensional (3D) data in the universal data format (ISO 10303) and comprises geometry data from CAD in Boundary representation (B-rep) [2]. The International Standards Organization (ISO) developed STEP (ISO 10303) as a universal standard to extend STEP to overcoming proprietary data restrictions between CNC machines [3].

CNC machines are high-tech machines that utilize specialized software to drive the machine's axial movement automatically. These CNC machines have become an integral part of the manufacturing and automation sector. This

*Corresponding author: saifulbahri@unisza.edu.my*

machine outperforms conventional mass-production machines in terms of precision and production rate. In manufacturing, this CNC is the main part of a machine tool that serves as the brain in controlling a machine's movement. It works with a G-code (ISO 6983), a low-level language that allows the machine application to manage and control the tool path movement in the programmed direction [4]. The G-code language is designed as a medium to transfer data between a computer and a CNC system. G-code bridges the CNC machine and controller to allow the machining process to be performed as programmed. The G-code works by instructing machine tools straight or curved toward a programmed set of coordinate points.

G-codes can be generated manually or automatically in the current manufacturing process. The machine operator will manually interpret the CAD pattern and generate G-codes based on existing design features in the CAD data model. Typically, such a method is prone to error and takes considerable time. G-code programs are created using CAM tools for automated generation methods [5]. However, this method is costly to implement, and CAM software typically only supports specific CAD design formats. CAM also necessitates a more in-depth understanding of the particular specifications for models, equipment, machine tools, peripherals, and software [6], [7]. Post-CAM processors will recognize programming in their own proprietary CNC code format, resulting in data format inconsistencies. While current CNC manufacturing technology is advancing rapidly, this incompatibility will further stymie advancements in CNC-based machining. Data redundancy and possibly significant errors will result from all of these constraints [8].

STEP enables the independent interpretation of data structures based on product models throughout the product's lifecycle. G-code is primarily used in computer-aided manufacturing systems to control automatic machine tools. Both provide standardized product data that can be exchanged and further integrated. As a possible consequence, data format incompatibilities and vendor requirements can be avoided. Furthermore, the introduction of Open Architecture Control (OAC) technology bridges this gap further. OAC technology intends to create vendor-neutral control system interfaces and specifications for operability, interaction, portability, and scalability [9].

The geometry product data in the STEP file must be carefully selected and translated for the geometry product chosen to be used in the format and structure of the G-Code program. Among other things, STEP provides geometric data and high-level data on machining characteristics, preparation, work plans, and work steps that can be used to advance to the next stage of production planning and control. The issue with STEP files is that their data structure is inefficient because it repeats across multiple rows and contains information that is not required to generate geometric profiles for machining operations. While STEP files are adaptable and universal, converting all data to G-code is not an easy task. Numerous events represent data in the STEP file, and each event represents a subset of the geometric data contained in the CAD model.

## 2. STEP File Data Structure

The STEP file format is an international standard for encoding design features for CAD models. STEP is a platform for converting CAD systems and software data models that support industry-standard CAD formats. STEP encourages representation and communication throughout the product's life cycle, from manufacturing to process planning, prototyping, and design. Additionally, STEP can be manipulated and used to optimize the exchange of product data, serving as a reference model for a comprehensive description of product data [10]. STEP is a high-level programming language that allows for creating solid models with CAD software. The model's data structure can provide a bidirectional data flow capable of channeling data between CAD, CAM, and CNC platforms without redundancy or risk of information loss [11]. The exchange data should be defined using object-oriented modeling methods. There should be no data leakage on the Computer-Aided technology (CAx) data chain [12].

Today's intelligent controllers are specifically designed to analyze and interpret STEP file information when generating, simulating, and optimizing machining tool paths [13]. The STEP file is universal, and its data structure is an open-source structure that is not specific to any machine or brand. It can be applied to various open-source controllers for most machine tools on the market [14]. When this STEP file is used as the input file for the machining process and is integrated with open-source control software, the need for a post-processor is addressed. Because the file contains sufficient information and data to create the tool path and generate G-code that indirectly performs the machining process [15].

STEP files are open-source extension files that can be accessed by any typical standard software package, including Notepad, Textfile, etc. Similar to the keyword (ISO-10303-21), the STEP file's end contains the string (END-ISO-10303-21), and the section is similarly described [16]. The STEP file's data structure consists of the HEADER and DATA sections. The HEADER section contains general information, including file title, author, and schema information. Events in the DATA section consist of solid model features, including close shell, lines, axes, units, etc. STEP files contain a wealth of solid feature information, which is why they can only be used to perform machining by generating CNC codes [17]. As a result, the STEP standard emerges as a viable option for extending CAx chain interoperability as CNC technology advances.

The STEP file structure is text-based universal data that can be opened, changed, edited, and saved in ASCII text format. The data structure includes grammar that is not dependent on the linguistic context and manipulating it can aid in developing applications. The STEP structure's grammar is specified in Wirth Syntax, and the file information is defined in a column-free format [18]. This file includes a B-rep file containing geometric data from 3D CAD models.

This geometric data can generate CNC machining G-codes if thoughtfully manipulated, extracted, filtered, eliminated, integrated, and decoded. STEP files contain data extracted repeatedly in a specific order, and understanding how to interpret STEP files requires knowledge of geometric data hierarchies. In general, not all of the information and data contained in the STEP file is necessary to execute the process for generating this tool path and G-code. Only information relating to cartesian coordinates and specific machining features is used.

## 3. Research Methodology and System Development

According to Figure 1, the methodology's specific goal is to convert STEP files into G-code, providing an alternative method for automatically generating G-code and facilitating data flow between the CAD and CAM environments. In short, the 3D solid model is created by the user or programmer using CAD modeling software as the product model and is stored in a STEP file format. This system will analyze and manipulate all geometric data in the STEP file before translating it into the algorithm developed using the Geometric Data Extraction (GDE) approach. The geometric data algorithm is created using a workflow diagram, which simplifies and facilitates data representation. During this process stage, the standard technique in determining a pattern for the hierarchical tree in the STEP file is to identify and analyze all of the STEP file data structures.

This algorithm was developed to extract geometric data and define data entities and designs by utilizing a B-rep database that contains geometric and topological information. The algorithm will interpret the data and structure contained in the B-rep data obtained by translating it to a hierarchical tree arrangement form. The Hypertext Preprocessor (PHP) software produced the specific interface system for extracting geometric data from STEP files from any part of the modeled workflow. With all data information, the path pattern of the profile, whether circular or linear, can be calculated and recognized.

The $I^2S$ is designed to accept user input and requires additional data to generate tool paths and CNC code, including the G/M code, cutting tool, feed rate, and spindle speed. This $I^2S$ system architecture includes specific data modules, such as read/write/rewrite, extraction, elimination, processing, integration, and transformation. During the read/write/rewrite module stage, the interface will analyze all data information in the STEP file line-by-line. Each analysis will yield the final result, which contains the cartesian point for x, y, and z representing the coordinate readings to generate the machining tool path and G-code [19]. The following STEP file will be implemented into the extraction module, analyzing each line. The interface will extract all of the information and STEP file geometric data and write and restructure it before recording it in the database.

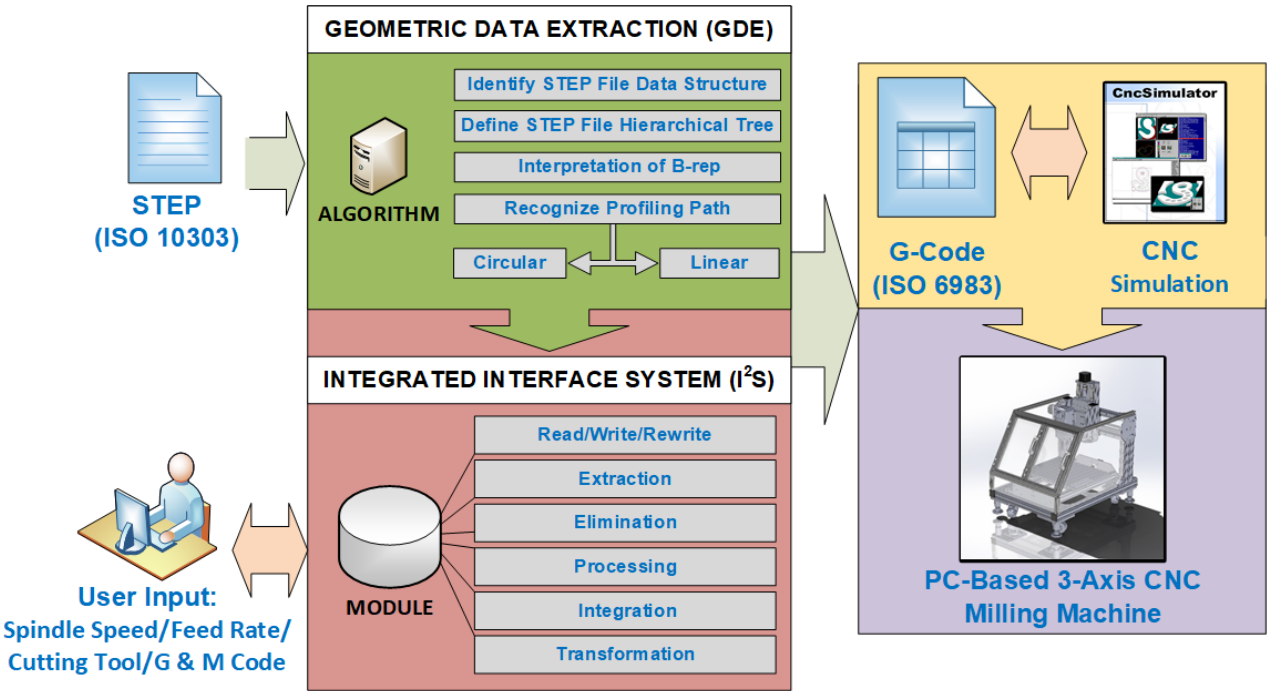



**Fig. 1 - Research methodology framework**

After that, all of the data in this database will be filtered by removing duplicate, redundant, and repetitive data that is unimportant or necessary in generating the machine tool path through the data elimination module. The repeating entity will be eliminated, and removing redundant data is vital in designing the interface. Because STEP files are B-rep in nature, many geometric entities will inevitably appear multiple times across STEP files. A single entity can overlap with another type of entity various times. The data is routed to the processing module following the extraction module, which uses defined and developed algorithms to process the entire data set. Several submodules of processing algorithms are dedicated to data extraction, elimination, processing, and integration. The next module is concerned with data integration between cutting parameters and geometric data. This module integrates user information about specific parameters such as spindle speed, feed rate, cutting tool, and G&M code, offering additional input for tool path generation. These cutting parameters will be integrated with the geometrical data provided by the STEP file.

The final module is the transformation module, which automatically generates a complete G-code generation process based on the input data. This process generates a movement sequence for each axis, transforms the data in the STEP file to a cutting tool path movement, and generates the G-code. The resulting G-code is validated using a PC-based open architecture controller, and a tool path simulation is tested using CNC Simulator software. The sample parts are milled using a 3-Axis CNC milling machine to allow point-to-point operation profile cutting processes during the machining process.

## 3.1 Interpretation of the Algorithm

Verification through a case study is an option for this paper to validate the workflow, algorithms, and methods developed. The hierarchical tree diagram can summarize and translate the algorithm development for the case study. In general, the STEP file contains data for boundary representation, geometric surface data, and geometric curves that serve as the basis for its formation. Shells, faces, loops, vertex, and edges are the entities used to represent data boundaries. Geometric surface data includes entities for planes and cylinders [20].

Meanwhile, the entities for circles and lines are contained within the data for geometric curves. STEP files' data structures are typically represented by rows of entities that do not appear in any particular logical order. Regarding feature geometry recognition, it is also unreadable by a programmer. Creating workflows and hierarchical trees is a practical way of reorganizing all of the data in a difficult-to-read STEP file that contains data that is not sequentially ordered line by line. The features recognition system and the STEP file shape typically started with the CLOSED_SHELL and ended with the CARTESIAN_POINT [21]. Other strings are also present, consisting of ADVANCE_FACE, FACE_BOUND, FACE_OUTER_BOUND, EDGE_LOOP, ORIENTED_EDGE, and so on [22]. They can also be identified and illustrated in a hierarchical structure tree, as shown in Figure 2.

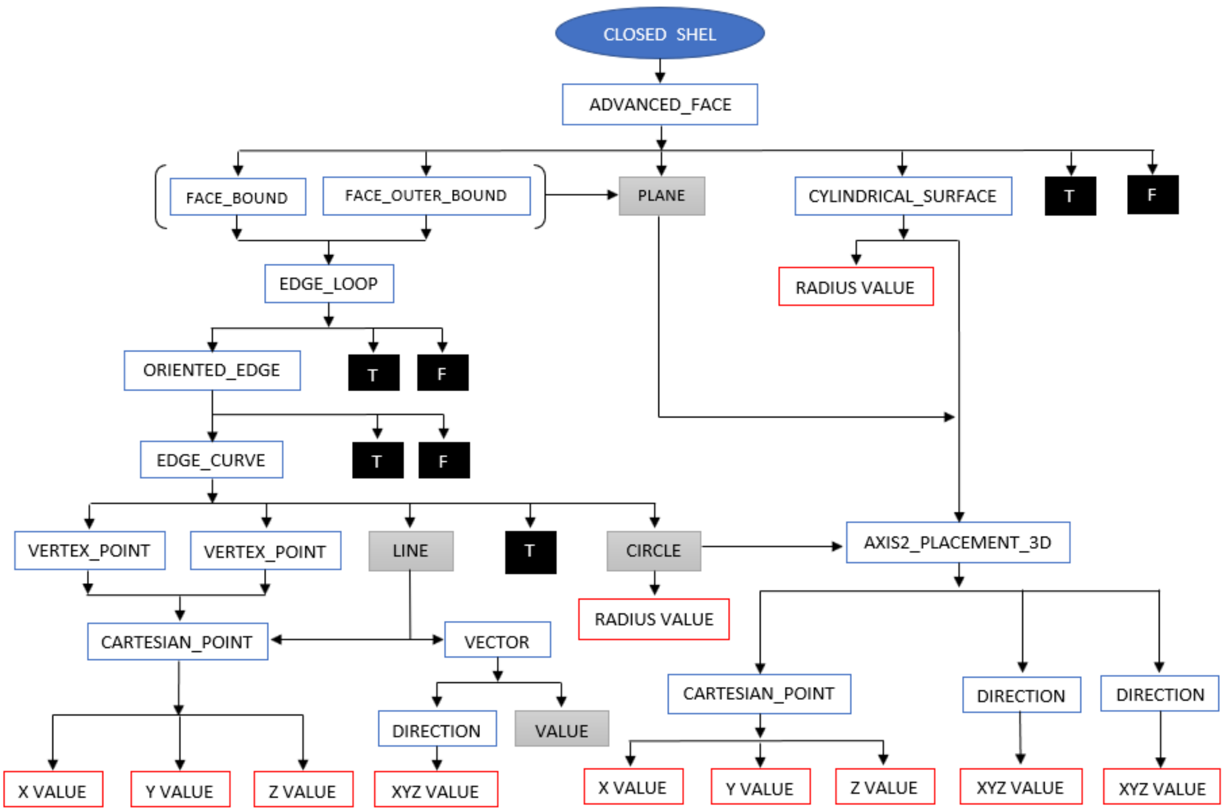



**Fig. 2 - Hierarchical tree of the STEP file structure**

Based on the stages of analysis and arrays in this hierarchical tree diagram, beginning with the determination of the entity value illustrated in Figure 3, the result for the EDGE_LOOP entity appear is #154 = EDGE_LOOP (#595, #649,

#379, #904, #82, #590, #363, #654, #375, #851). The presence of EDGE_LOOP entities in this 3D object includes correlations and combinations for all other entities such as ORIENTED_EDGE, EDGE_CURVE, VERTEX_POINT, and CARTESIAN_POINT. Finding all of the EDGE_LOOPS presented for a sample is the primary issue, and errors in determining the correct EDGE_LOOP will result in system errors and failure.

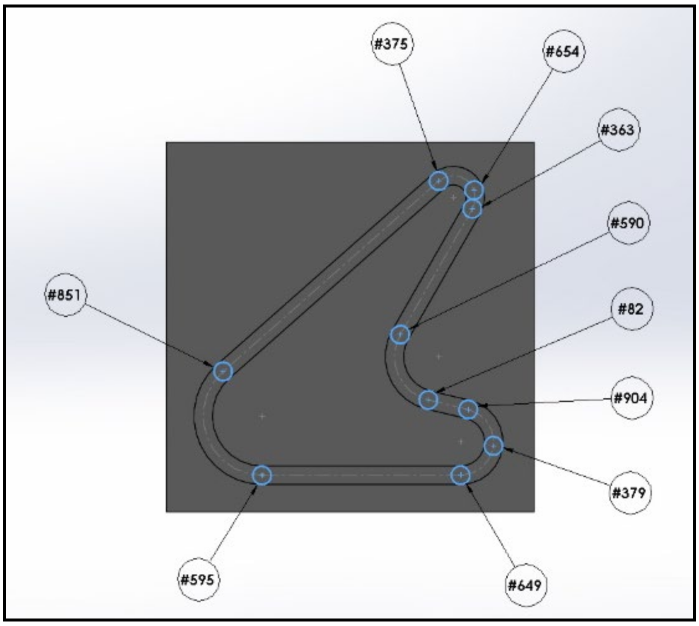
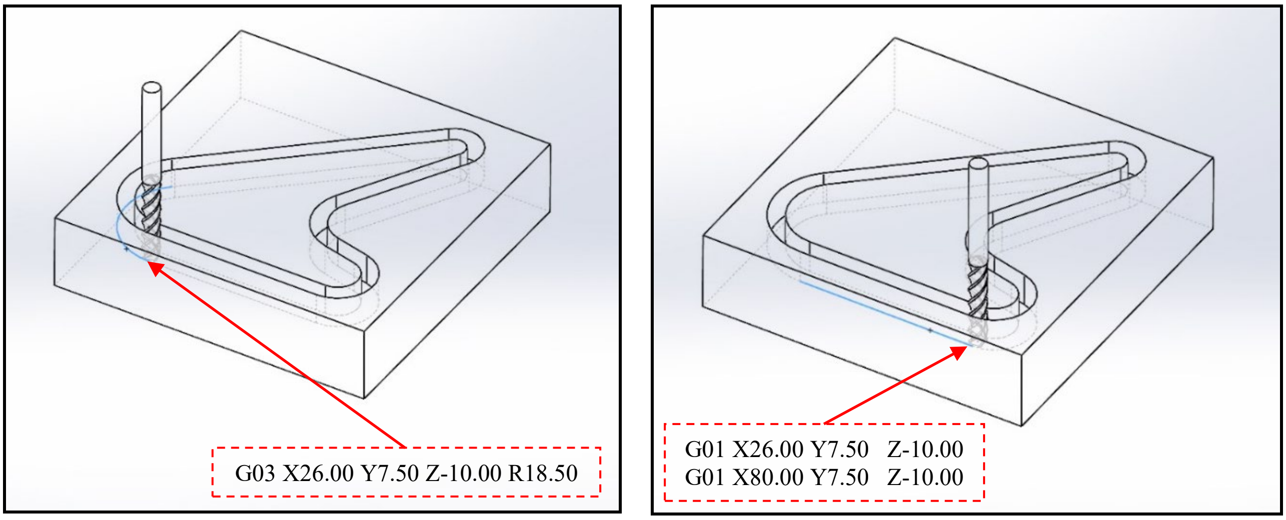



**Fig. 3 - The entity #154 in the EDGE_LOOP array**

As shown in Figure 5, describe geometric data extraction in the hierarchical tree of the entity EDGE_LOOP (#595). Once the EDGE_LOOP entity value #154 is analyzed, it will link to #595, the first value in brackets. This value is extracted and processed sequentially to determine the value of the ORIENTED_EDGE. Regarding that, selecting values for EDGE_CURVE (#645) commences determining values for VERTEX_POINT (#209, #632) and CIRCLE (#502). Two values for the CARTESIAN_POINT (#605, #688) will be generated from the VERTEX_POINT value, which is valuable for deciding on the start point and the value used to determine the endpoint. At the same time, the value of this CIRCLE will yield two values, namely the determination of AXIS2_PLACEMENT_3D (#92) and the radius value. The result for the value of AXIS2_PLACEMENT _3D will produce three other values to identify the value of CARTESIAN_POINT (#349), representing the value of the center of the arc.




First Point Second Point

**Fig. 4 - The cutter's position**

While the two different values are used to determine the DIRECTION (#673, #701) in which the tool path will move, either clockwise or counterclockwise. The results obtained in this case show that the result readings for these two directions indicate counterclockwise movement (G03), with positive reading directions at the x, y, and z axes and negative reading directions at the x-axis. Following this procedure, the output of the result is G00 X13.79; Y39.90 Z10; G01 X13.79 Y39.90 Z-10 and G03 X26.00 Y7.50 Z-10 R18.50. The cutter will move from the starting point G01 X13.79 Y39.90 Z-10 to the second point G03 X26.00 Y7.50 Z-10.00 R18.50. The cutting tool's movement is illustrated in Figure 4 in the first point position, where it mills the workpiece in a circular motion in the counterclockwise (G03) direction with a radius of 18.50.

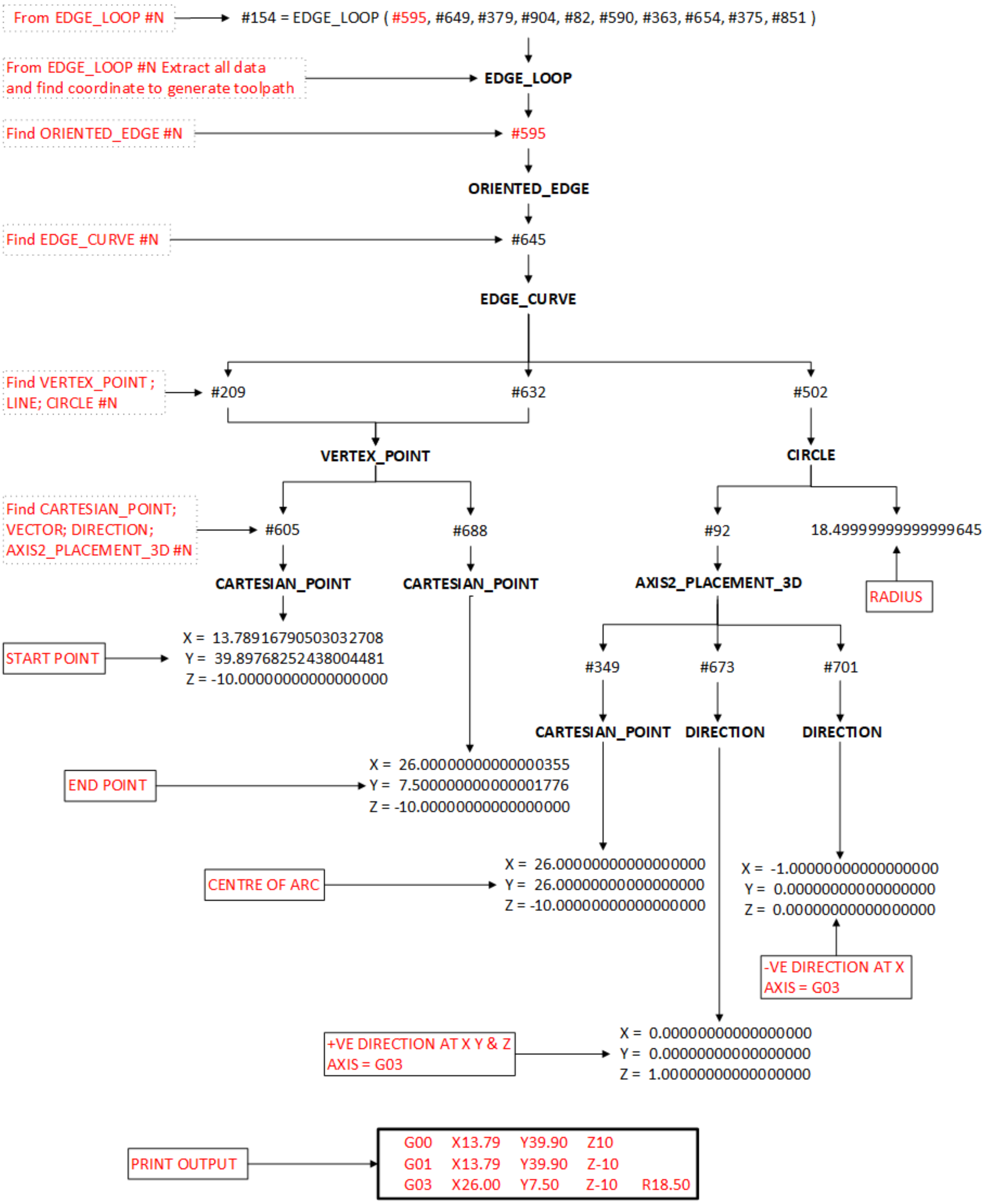



**Fig. 5 - Hierarchical tree of the EDGE_LOOP (#595) for circular cutting**

Meanwhile, the cutting tool's movement for the next position is at #649. These values are extracted and processed sequentially to identify simultaneous values such as ORIENTED_EDGE (#649), EDGE_CURVE (#53), VERTEX_POINT (#632, #22), LINE (#319), and ending with CARTESIAN_POINT (#688, #88) for x, y, and z. The value of VERTEX_POINT will generate two values for CARTESIAN_POINT: the start points and the endpoint. In contrast, the value of this LINE will produce two different values: the CARTESIAN_POINT (#726) and the VECTOR (#612). At the same time, this CARTESIAN_POINT value will yield a value equal to the value used to determine the endpoint. However, the VECTOR value produces two values, one of which is the DIRECTION (#720), which refers to the movement of the tool path and produces a positive value on the x and y axes.

Besides that, the vector produces a value of 1000.000000000000, which is also the reading result for the straight line. The results obtained in this case show that the result readings for these two directions show movement in a straight line (G01). Following this process, the output is G01 X26.00 Y7.50 Z-10 and G01 X80.00 Y7.50 Z-10. Additionally, the entire process will refer to the illustration in the following hierarchical tree in Figure 6. The G-code program generated from this hierarchical tree is shown in Figure 4 in the second point position. They can also be translated into a graphical representation depicting the position of the end mill at the starting point, G01 X26.00 Y7.50 Z-10.00, and moving in a straight line to the second point, G01 X80.00 Y7.50 Z-10.00, with the same depth as Z-10.

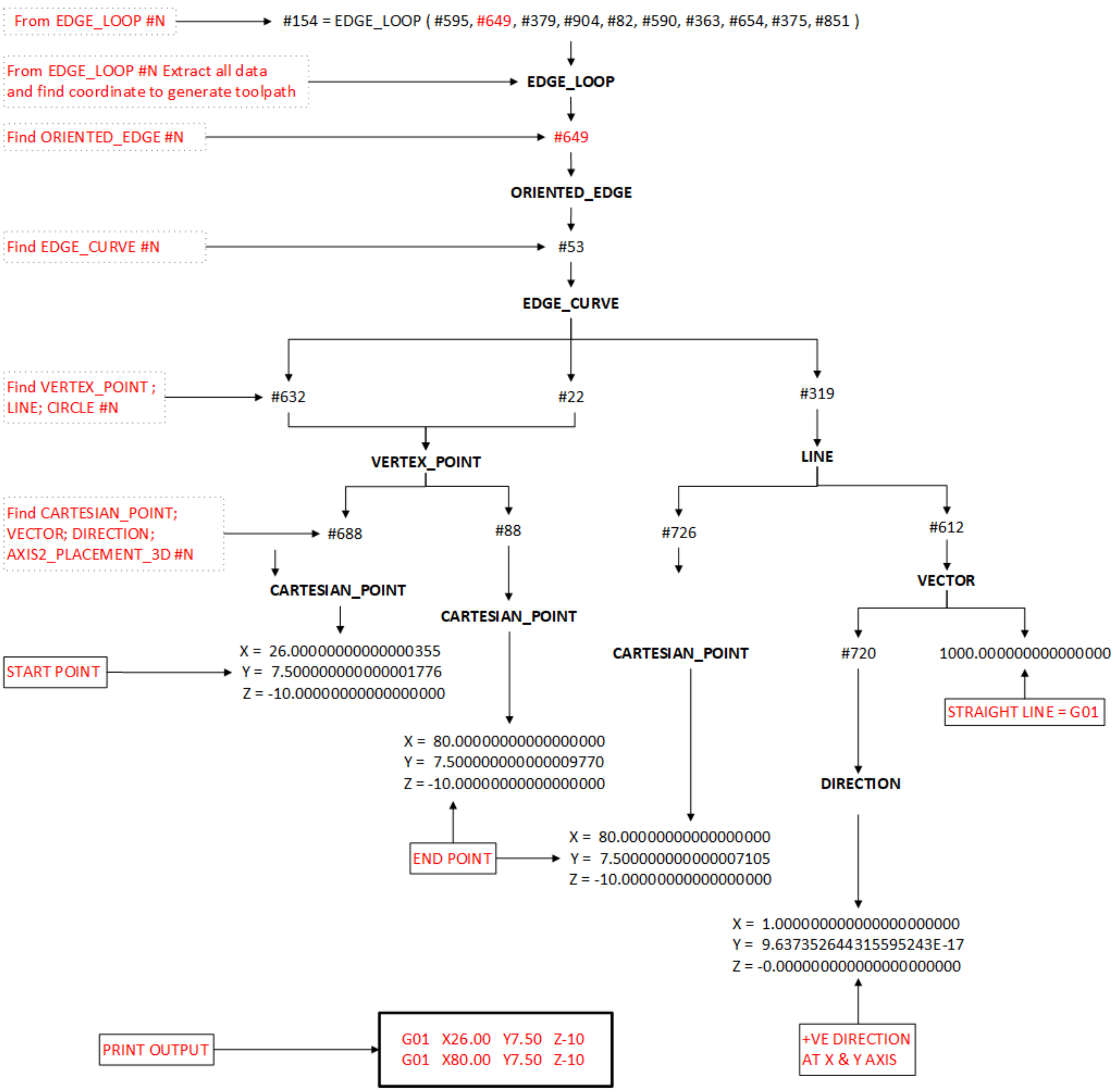
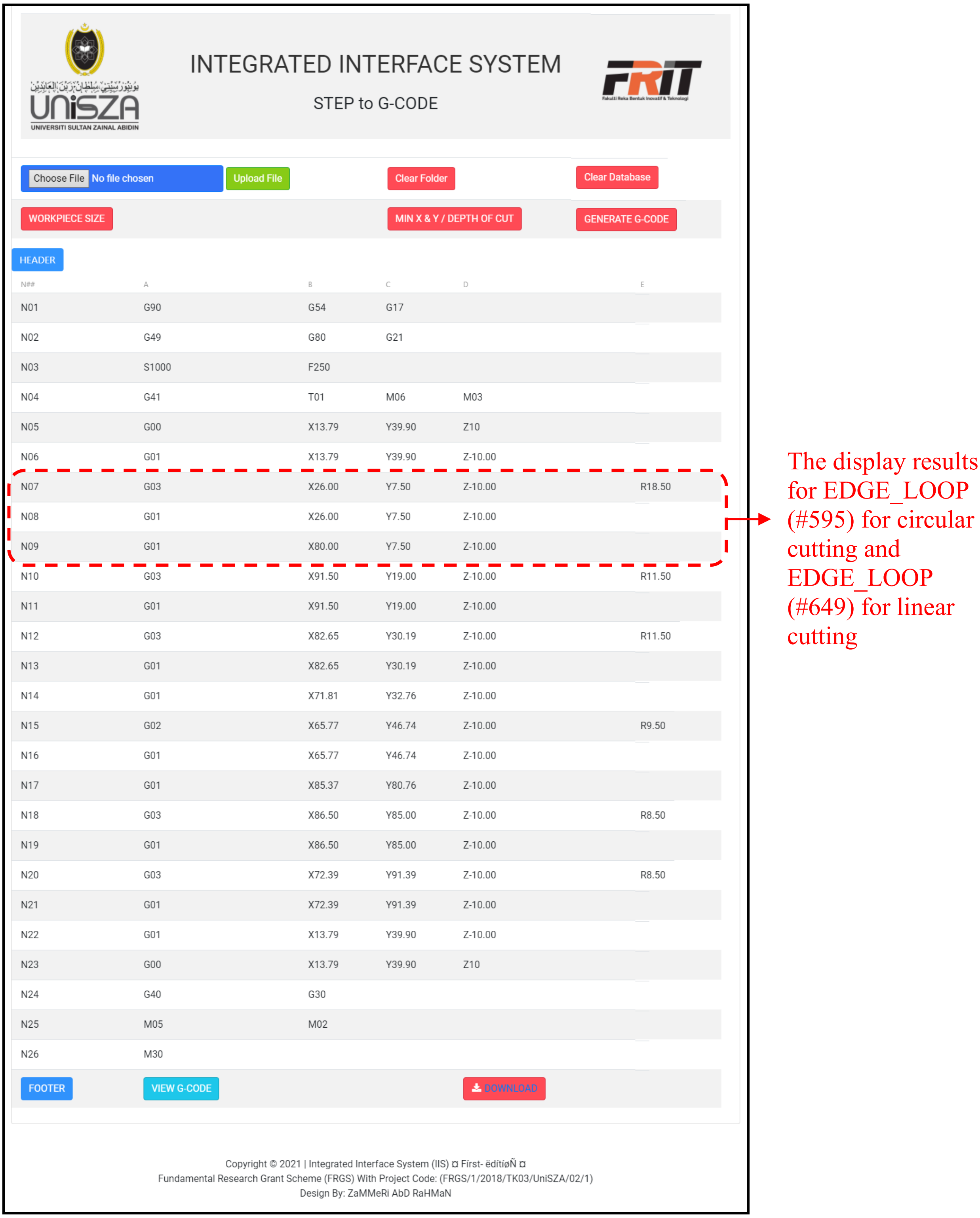



**Fig. 6 - Hierarchical tree of the EDGE_LOOP (#649) for linear cutting**

## 4. The Result of the Case Study

The model sample for these case studies was created using Ultra High Molecular Weight Polyethylene (HWPDE) as the working material, and the solid 3D models were created using Solidworks software. The hierarchical tree and algorithms were developed to structure the architecture of the I²S system to achieve the desired output capability. After the analysis and process of developing algorithms and methods succeeded implemented, the I²S will be tested using the samples that have been created. This process is carried out in stages to test the functionality of the I²S programs and systems. Figure 7 depicts the output of the auto-generate G-code for the sample study.

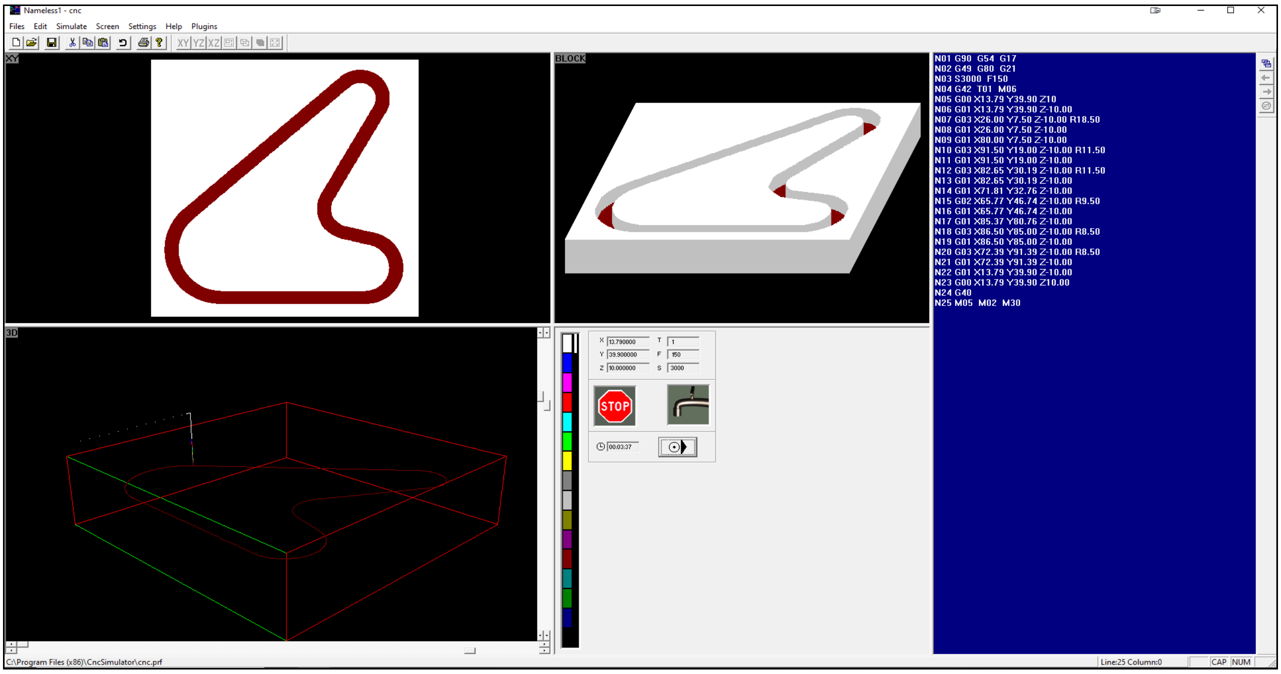



| N## | A | B | C | D | E |
|---|---|---|---|---|---|
| N01 | G90 | G54 | G17 | | |
| N02 | G49 | G80 | G21 | | |
| N03 | S1000 | F250 | | | |
| N04 | G41 | T01 | M06 | M03 | |
| N05 | G00 | X13.79 | Y39.90 | Z10 | |
| N06 | G01 | X13.79 | Y39.90 | Z-10.00 | |
| N07 | G03 | X26.00 | Y7.50 | Z-10.00 | R18.50 |
| N08 | G01 | X26.00 | Y7.50 | Z-10.00 | |
| N09 | G01 | X80.00 | Y7.50 | Z-10.00 | |
| N10 | G03 | X91.50 | Y19.00 | Z-10.00 | R11.50 |
| N11 | G01 | X91.50 | Y19.00 | Z-10.00 | |
| N12 | G03 | X82.65 | Y30.19 | Z-10.00 | R11.50 |
| N13 | G01 | X82.65 | Y30.19 | Z-10.00 | |
| N14 | G01 | X71.81 | Y32.76 | Z-10.00 | |
| N15 | G02 | X65.77 | Y46.74 | Z-10.00 | R9.50 |
| N16 | G01 | X65.77 | Y46.74 | Z-10.00 | |
| N17 | G01 | X85.37 | Y80.76 | Z-10.00 | |
| N18 | G03 | X86.50 | Y85.00 | Z-10.00 | R8.50 |
| N19 | G01 | X86.50 | Y85.00 | Z-10.00 | |
| N20 | G03 | X72.39 | Y91.39 | Z-10.00 | R8.50 |
| N21 | G01 | X72.39 | Y91.39 | Z-10.00 | |
| N22 | G01 | X13.79 | Y39.90 | Z-10.00 | |
| N23 | G00 | X13.79 | Y39.90 | Z10 | |
| N24 | G40 | G30 | | | |
| N25 | M05 | M02 | | | |
| N26 | M30 | | | | |

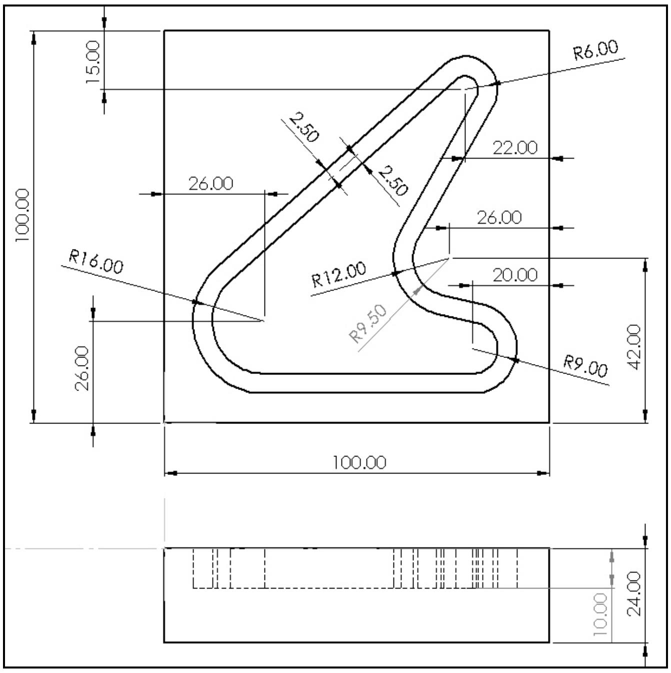
**Fig. 7 – The result of the auto-generate G-code**

## 4.1 Result Validation

Case studies are one approach for validating the workflow, algorithms, and methods developed in this study for generating tool paths and G-code programs. Figure 8 depicts the simulation results, CAD design feature, and machining output comparison. The first image represents the simulation process, a screen display of the cutting tool's movement, and an object representation in the form of a solid 3D model composed of CncSimulator software-generated circular and linear tool movements. The following image shows a 3D technical drawing of a model sample designed to fulfill its specific specifications and dimensions.




Tool-path simulation in CNC simulator




3D CAD Model

Physical Machining

**Fig. 8 - Comparison result between CNC simulation, CAD design feature, and physical machining**

Meanwhile, the figure also describes the machining of Ultra-High Molecular Weight Polyethylene (UHMWPE) workpieces using a bench-top three-axis CNC milling machine controlled by open architecture control software. This physical CNC machining will compare the initial 3D CAD design, simulation results, and final product. Finally, it is possible to conclude and prove that the initial CAD drawings, simulation results, and finished products all have the same profile shape and size. Additionally, it is reasonable to infer that the interface system can successfully generate G-code through the geometric data extraction process.

## 5. Conclusions

The purpose of this article is to describe the design and development of an integrated interface system capable of extracting geometric data from STEP files and converting it to G-code for CNC machining operations. This interface can identify geometric features, mainly circular and linear, by analyzing the STEP data structure and simplifying it using the geometric data extraction method. A hierarchical tree is constructed to define all data and entities present in the STEP file to translate it to the machining tool path. The algorithms developed in this study were written in the PHP programming language and were executed on a computer. In a nutshell, the experiment introduced a new system capable of converting STEP files to cutting tool movements in manufacturing operations. A critical concept in CAD/CAM integration is using a STEP-neutral file as a drive constraint in CNC programming and inputting the machining process's operation.

Additionally, the study recommends a procedure for determining sample results from CNC simulations and physical machining and profile features on CAD designs. CNC machining and simulation validate the steps necessary to develop an integrated interface system. When replicated using a CAD model, this test yielded similar results. The apparent consistency of the results indicates that the STEP data structure and workflow construction used in this research is valid. The finding is that this developed program can be used as an alternative to converter programming based on ISO 6983 standards, specifically for STEP to G-code conversion. This method can also integrate data between CAD and CAM environments, which can help to expedite and reduce the product development cycle indirectly. Furthermore, for future work, this intelligent interface system can be designed to enable advanced systems to combine complex machining features and generate more optimal machining tool paths by taking specific data into accounts, such as machining parameters, strategies, and functions.

## Acknowledgement

The authors express the most profound appreciation to University Sultan Zainal Abidin (UniSZA), in particular to University Teknikal Malaysia Melaka (UTeM), University College TATI (UC TATI), and especially to the Ministry of Higher Education (MOHE) Malaysia for financial and granting funding to this research under the Fundamental Research Grant Scheme (FRGS) (Grant No. FRGS/1/2018/TK03/UNISZA/02/1).

## References

[1] J. Ismail, S. B. Mohamed, A. Mohd, and Z. Abd Rahman, "Tool Path Generation for Linear Cutting Using Step File ( ISO 10303 ) Data Structure," *International Journal of Advanced Research in Technology and Innovation*, vol. 3, no. 4, pp. 53–64, 2021, [Online]. Available: http://myjms.mohe.gov.my/index.php/ijarti.

[2] Y. Qin, Q. Qi, P. J. Scott, and X. Jiang, "Status, comparison, and future of the representations of additive manufacturing data," *CAD Computer Aided Design*, vol. 111, pp. 44–64, 2019, doi: 10.1016/j.cad.2019.02.004.

[3] T. M. S. Tengku Sulaiman, S. B. Mohamed, M. Minhat, A. S. Mohamed, A. R. Mohamed, and S. N. A. Yusof, "File and PC-Based CNC Controller using Integrated Interface System (I2S)," *Journal of Advanced Research in Applied Mechanics*, vol. 70, no. 1, pp. 1–8, 2020, doi: 10.37934/aram.70.1.18.

[4] K. Latif, Y. Yusof, and A. Z. A. Kadir, "Development of new integrated CNC system for ISO 6983 data interface model," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 2021, doi: 10.1007/s12008-021-00837-2.

[5] R. B. Käsemodel, A. F. de Souza, R. Voigt, I. Basso, and A. R. Rodrigues, "CAD/CAM interfaced algorithm reduces cutting force, roughness, and machining time in free-form milling," *The International Journal of Advanced Manufacturing Technology volume*, vol. 107, pp. 1883–1900, 2020, doi: 10.1007/s00170-020-05143-x.

[6] S. Rahimić and A. Memić, "Intelligent CAD Systems for Generation G Code," in *International Conference "New Technologies, Development and Applications" (NT 2020): New Technologies, Development and Application III*, 2020, pp. 190–193, doi: 10.1007/978-3-030-46817-0_21.

[7] T. B. Tuli and A. Cesarini, "Automated unsupervised 3D tool-path generation using stacked 2D image processing technique," *Journal of Manufacturing and Materials Processing*, vol. 3, no. 4, 2019, doi: 10.3390/jmmp3040084.

[8] S. H. M. Said and M. S. A. Mansor, "A Study on Post Processor for 5-Axis CNC Milling via CAD/CAM System," in *Symposium on Intelligent Manufacturing and Mechatronics (SympoSIMM 2021): Intelligent Manufacturing and Mechatronics*, 2022, pp. 448–456, doi: 10.1007/978-981-16-8954-3_42.

[9] L. Kamran, "New Technique for the Development of Open Cell Controller based on ISO 14649 and ISO 6983,"

Universiti Tun Hussein Onn Malaysia, 2015.
[10] 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*, vol. 104, no. 1–4, pp. 261–284, 2019, doi: 10.1007/s00170-019-03849-1.
[11] Y. Liu, G. Zhao, O. Zavalnyi, X. Cao, C. Kang, and W. Xiao, “STEP-Compliant CAD/CNC System for Feature-Oriented Machining,” *Computer-Aided Design & Applications*, vol. 16, no. 2, pp. 358–368, 2019, doi: 10.14733/cadaps.2019.358-368.
[12] J. Vukman, S. Zivanovic, and M. Milosevic, “Application of STEP-NC in the integration CAX and CNC systems: Case studies,” no. January, 2016.
[13] M. A. Othman, Z. Jamaludin, and M. Minhat, “Intelligent Control of CNC System Based on IEC 61499 Function Block Technology,” in *Symposium on Intelligent Manufacturing and Mechatronics (SympoSIMM 2019): Intelligent Manufacturing and Mechatronics*, 2019, pp. 176–185.
[14] S. B. Mohamed *et al.*, “Integrated interface system for tool path generation of STEP file,” *AIP Conference Proceedings*, vol. 2129, no. August, 2019, doi: 10.1063/1.5118020.
[15] P. Hu, Z. Han, H. Fu, and D. Han, “Architecture and implementation of closed-loop machining system based on open STEP-NC controller,” *Architecture and implementation of closed-loop machining system based on open STEP-NC controller*, vol. 83, no. 5–8, pp. 1361–1375, 2016, doi: 10.1007/s00170-015-7631-z.
[16] K. Latif, A. Adam, Y. Yusof, and A. Z. A. Kadir, “A review of G code, STEP, STEP-NC, and open architecture control technologies based embedded CNC systems,” *International Journal Advance Manufacturing Technology*, vol. 114, pp. 2549–2566, 2021, doi: 10.1007/s00170-021-06741-z.
[17] 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*, vol. 7, no. 2.15, p. 27, 2018, doi: 10.14419/ijet.v7i2.15.11194.
[18] 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*, vol. 32, no. 2, pp. 219–231, 1996.
[19] 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*, vol. 12, no. 7, pp. 195–203, 2020, doi: 10.30880/ijie.2020.12.07.022.
[20] P. B. Mangesh and N. Rakesh, “STEP-based feature extraction from STEP geometry for Agile Manufacturing,” *Computers in Industry*, vol. 41, no. 1, pp. 3–24, 2000, doi: 10.1016/S0166-3615(99)00040-8.
[21] 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*, vol. 3, no. 6, pp. 102–115, 2012, doi: http://dx.doi.org/10.4314/ijest.v3i6.9.
[22] X. Huang, G. Zhou, and R. Liang, “A Web 3D Viewer Based on STEP,” *16th International Conference*, 2007, doi: https://doi.org/10.1109/ICAT.2006.30.