Indonesian Journal of Electrical Engineering and Computer Science
Vol. 31, No. 3, September 2023, pp. 1362~1371
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v31.i3.pp1362-1371

G-code converter using interface system for a STEP file (ISO 10303)

Zammeri Abd Rahman[1], Saiful Bahri Mohamed[1], Mohamad Minhat[2], Alias Mohd[3], Zulkifli Abd Rahman[3], Roslan Awang[1]
[1]Faculty of Innovative Design and Technology, Universiti Sultan Zainal Abidin (UniSZA), Kuala Nerus, Malaysia
[2]Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Malaysia
[3]Faculty of Engineering Technology, Universiti College TATI (UC TATI), Kemaman, Malaysia

Article Info

Article history:

Received Jan 19, 2022
Revised Jun 4, 2023
Accepted Jun 17, 2023

Keywords:

Computer numerical control
G-code
Interface
Milling machine
STEP

ABSTRACT

The STEP file is the "standard for the exchange of product model data," which is usually used to exchange geometric data in boundary representation (B-rep) between different computer-aided design (CAD) platforms. These data can be fully utilized and integrated into a larger manufacturing organization, such as a computer-aided manufacturing (CAM) environment for computer numerical control (CNC) machining applications based on ISO 6983. The 3D models of 2D machining profiles were created in CAD software and saved as STEP files. The data structure was analyzed by comparing the geometric entities of the CAD model and the STEP file. The algorithm was created using the hypertext preprocessor (PHP) programming language and produced a computer interface system to convert STEP files into G-code format. The machining blocks with profile machining features were simulated using a CNC simulator and PC-based open architecture control (OAC) software. The G-code was validated on a three-axis CNC milling machine, and the result was compared to the CAD model to confirm the machining profile. The integrated interface system I^{2}S demonstrates its ability to interpret all 2D profiles and generate machining tool paths and G-code, allowing data flow between CAD and CAM environments and shortening product development cycles.

Corresponding Author:

Saiful Bahri Mohamed
Faculty of Innovative Design and Technology, Universiti Sultan Zainal Abidin (UniSZA)
Gong Badak Campus, 21300 Kuala Nerus, Terengganu, Malaysia
Email: saifulbahri@unisza.edu.my

1. INTRODUCTION

Computer numerical control (CNC) is a system that employs a combination of computer-aided design (CAD), computer-aided process planning (CAPP), computer-aided manufacturing (CAM), and CNC itself, collectively known as computer-aided technologies (CAx) chain. This integration enables the automation of machining processes from design to production, resulting in enhanced efficiency, cost savings, and improved product quality [1]. With the widespread adoption of CAD/CAM, most manufacturing industries now possess multiple CNC machines to accommodate business growth and meet customer demands. Operators of CNC machines have the flexibility to design and customize machine parameters such as feed rate, cutting depth, spindle speed, and more. Modern CNC machines offer advanced capabilities such as controlling multiple axes, error correction, and performing diverse tasks [2].

G-code is the programming language for CNC machines and is commonly used to control them through CAD/CAM software. Its usage has become prevalent across various industrial sectors worldwide [3]. The CAD-CAM-CNC approach has been widely employed in manufacturing for a considerable period [4]. The following passage describes different methods of generating G-code, a language used to control CNC machines. One approach involves manually generating G-code from a product drawing, which requires specialized skills and experienced employees. However, this process is time-consuming and prone to errors [5]. Another method utilizes CAM software to automatically convert CAD drawings into G-code [6]. This CAD-CAM-CNC chain streamlines the manufacturing process and eliminates the need for manual coding.

However, CAM software can be expensive, restrict users to specific software ecosystems, and potentially cause compatibility issues [7]. The third method involves converter software that can convert various file formats into CNC codes suitable for milling or laser machines. This converter supports multiple file formats, including image, standard template library (STL), object (OBJ), text, scalable vector graphics (SVG), drawing exchange format (DXF). It can automatically convert 3D file designs into G-code. It is worth noting that there is a lack of dedicated software specifically for converting STEP files to G-code, particularly for milling purposes.

In recent years, CAD/CAM systems have significantly progressed, incorporating features like stress and strain analysis on solid models and three-dimensional design simulations. However, sharing CAD model data among different platforms has led to duplication and fragmentation of the data system. The complexity of designing data networks within CAD systems has contributed to this issue [8], [9]. To address this challenge, the ISO 10303 STEP format has emerged as a standardized data format for technical specifications of machining parts. STEP is an open-source format where data is converted to G-code before being saved as a STEP file, allowing CAD software systems to extract relevant information [10], [11]. Preprocessors in CAD/CAM software packages widely utilize STEP-data encoding and decoding. STEP is a file format for exchanging CAD data and applies explicitly to 3D drawings [12]–[15].

Standard for the exchange of product model data (STEP) can generate G-code for CNC machining, a low-level programming language instructing machines on tool path movement. Previously, CNC machine manufacturers used no unified standard for encoding CAD designs before creating the STEP file. Post processors on CNC machines are responsible for decoding and encoding CAD models into G-code. Incompatibilities arise when machines adhere to different standards, leading to the additional time required for engineers to ensure proper functionality and potential production delays [16], [17]. Meanwhile, an integrated interface system has been developed to generate G-code directly from STEP files without relying on CAM software. This interface primarily reads STEP files generated by CAD software, which are computer-generated images of machined blocks derived from blank workpieces. The interface processes the STEP file, extracting geometric data to determine tool paths and generate G-code after scanning [18]. By reading and processing all instances within the STEP file, only the necessary information for programming output is extracted. The data flow of the integrated interface system is illustrated in Figure 1.

Figure 1. Data flow in integrated interface system (I^2S)

Over the past 50 years, machine tools have undergone remarkable advancements, transitioning from simple machines with memoryless controllers operated by perforated tape drives to sophisticated CNC multi-process workstations. These workstations possess various features like error compensation, multi-axis control, adaptive control, and the ability to integrate multiple manufacturing processes such as turn/mill/laser hybrid machines and grinding machines. However, the complexity of programming these advanced machines has become increasingly challenging. To address this complexity, CAD/CAM systems are offline development tools for generating and validating manufacturing processes [19]. While machine tools have undergone significant transformations, the fundamental programming language, G-code, has remained unchanged. G-code, or geometric code, is considered a low-level language primarily focusing on tooling, machine descriptions, and machine status. It provides CNC machines with geometric instructions, machining methods, and strategies. There have been revolutionary advancements in the generation of G-code from engineering drawings. However, these methods often require highly skilled individuals with extensive expertise to generate G-code manually. With the rise of technology, post-processors have emerged as a solution for automating G-code generation. Using STEP files as a neutral format for exchanging product data has made a significant and impressive contribution to advancing industrial processes and manufacturing technology. G-code remains the primary control language for all CNC machine tools [20]. Based on research findings, new concepts, and approaches have been developed to enhance the CNC machining process. Figure 2 illustrates a new method for the CNC machining process [21].

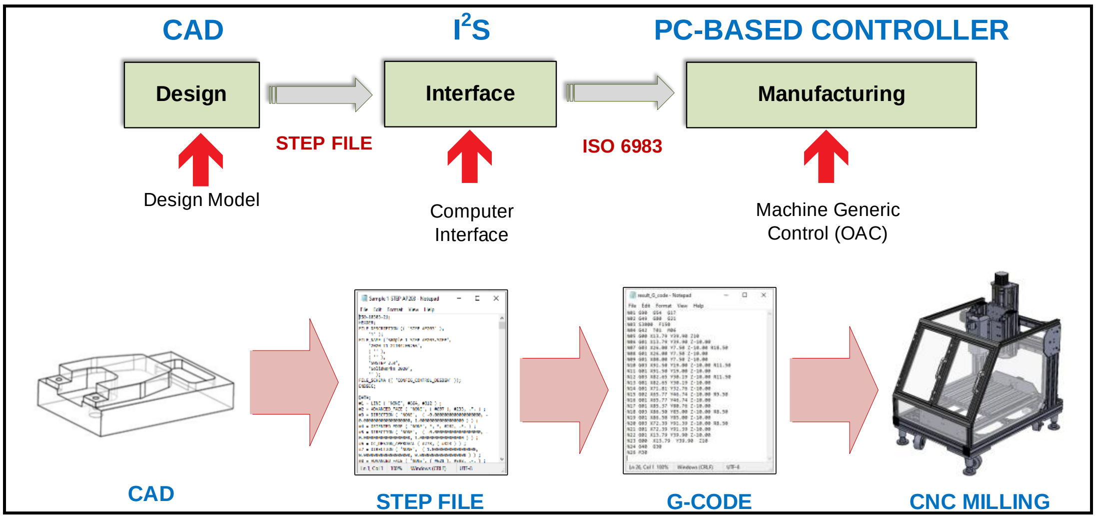

Figure 2. A new approach to CNC machining operation

2. RELATED WORK

Several free software programs, including freeware and shareware, have been developed by previous programmers and recognized as accessible tools for generating G-code. These third-party software solutions are designed to convert various language formats into the specific programming language used by CNC, 3D, milling, laser machines, and more. They support multiple file formats, such as image, STL, OBJ, Text, SVG, DXF, and others, allowing users to import and convert their preferred formats into G-code. These software tools can automatically convert 3D file designs into G-code, except for the lack of dedicated programs for converting STEP files to G-code. Figure 3 depicts a program created using both manual G-code generation and automatic G-code creation methods.

Although STEP converter-related software is available, they are primarily focused on converting to other formats and have not fully developed the capability to convert and generate G-code, particularly for milling operations. Limited previous research has been conducted in this area, and even if some studies exist, there is a lack of implementation specifically for STEP to G-code conversion. However, the STEP-compliant CNC systems (STEP-NC) field has been the research subject for many years. These systems incorporate intelligent functions that are impossible with conventional CNC coding operations (ISO 6983). STEP-NC has been developed through various research projects. It is important to note that while STEP-NC was created due to research efforts, it does not utilize the G-code function during machining operations and instead bypasses it entirely.

2.1. STEP file converter software

The development of the interface system plays a crucial role in establishing standardized machining processes for machine controllers in the CAx chain and facilitating operational manufacturing procedures. This interface system presents excellent opportunities for the advancement and growth of machining technology. It effectively leverages various techniques to extract data information from STEP files. An example of such an interface is the hole recognition system (HRS) developed by Tan *et al*. [22], which effectively identifies and extracts data related to multiple holes in a STEP file. However, it would have been even more advantageous if this system included a module for automatically converting STEP files into G-codes, incorporating additional features beyond hole recognition alone.

A revised version of the previous interface system showcased its potential in integrating the machining process with CNC machine tools by generating tool paths without relying on the CAM process. Nevertheless, this research had limitations as it primarily focused on linear features [23], [24]. To address this, Sharizam *et al*. [18] proposed a research approach to generate tool paths in the form of curves. The researcher argues that their approach eliminates the reliance solely on CAM software to generate G-code, instead utilizing source or proprietary software to assess CAD models and produce G-code. This eliminates the need for investing in CAM software and its associated high setup costs. Furthermore, these applications have employed PHP software to create visually appealing physical structures, mimicking the appearance and functionality of similar software on local PC.

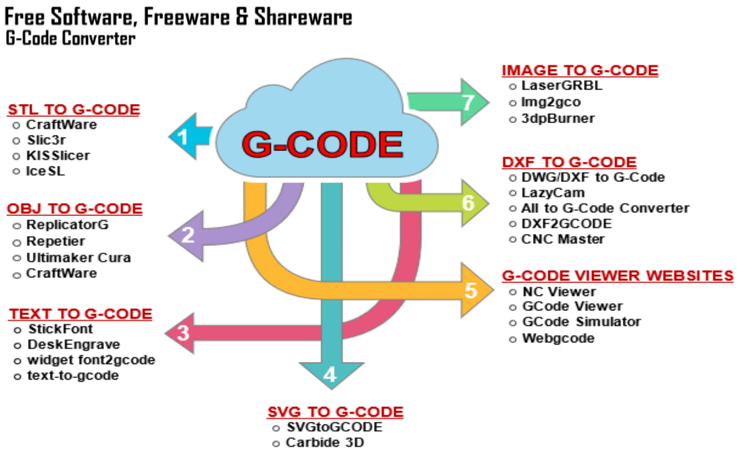

Figure 3. G-Code converter software

3. METHOD AND DEVELOPMENT

Previous research has focused on suggesting and developing new methodologies for creating integrated interface systems. One successful approach involved manipulating, understanding, and extracting STEP files, accomplished through developing a STEP file parser. This parser identified specific geometric features by employing generic methodologies and techniques, such as geometric data processing, geometric data extraction (GDE) method implementation, workflow construction, hierarchical tree creation, and algorithm development [25]. Before designing the interface, it is necessary to study and comprehend the machining feature present in a 3D CAD model and the structure of STEP data. The investigation analyzed how STEP files encode geometric data related to cartesian points, vertices, edges, and surfaces within a CAD system's solid model. The STEP file structure follows a language-based format with context-free grammar, expressed in the Wirth Syntax notation. The ISO 10303-21 STEP file format extension is an open-source file type (STP) that can be accessed using standard software applications like Notepad, Textfile, Word, and others. Typically, these files begin with the keyword "ISO-10303-21" and end with "END-ISO-10303-21."

3.1. Developing the integrated interface system (I^{2}S)

The development of the integrated interface system began after finalizing the research methodology workflow. This system manipulates geometric data from STEP files and involves several stages of modules: read/write/rewrite, extraction, elimination, processing, integration, and transformation. The primary objective of this development is to generate G-code from input data in the form of STEP files. The generated G-code is a general type that can be used with any open-source CNC controller. In this case, Mach4 is the CNC machine controller, a user-friendly open-source option available for free download. The interface system is developed using PHP, a widely-used open-source scripting language. Apache serves as the primary platform for building this interface system, and XAMPP control panel software has been employed to assist in its creation.

This software comprises various modules such as Apache, MySQL, FileZilla, Mercury, and Tomcat. To ensure the interface system's functionality, it underwent testing using STEP AP203 files. The generated G-code was thoroughly inspected to verify its adherence to the expected output. If any issues were identified, a debugging process was undertaken to address potential gaps and flaws. Once the G-code was deemed complete and ready, it underwent validation through simulation and physical machining on a CNC milling machine.

3.2. Interpretation of algorithm

Cluster B refers to the reading of the EDGE_LOOP path, which is crucial in determining the final G-code generation, as shown in Figure 4. The process involves extracting data step by step, following a top-to-bottom approach, to define the shape of the profile path for curved paths and the profiling path for straight lines. The output is organized in rows ordered by N and contains G-code commands such as G00, G01, G02, or G03, along with CARTESIAN_POINT coordinates specifying the x and y axes and the z-axis representing the cutting depth. G-codes G02 and G03 are significant as they indicate the tool path's movement along an arc, with the radius specified as 'r' in the G-code display. The extraction process involves identifying the presence of the EDGE_LOOP entity and retrieving all the related #N entities and geometric data. This process requires defining and interconnecting entities such as ORIENTED_EDGE, EDGE_CURVE, VERTEX_POINT, LINE, and CIRCLE to extract the necessary information and generate the tool path.

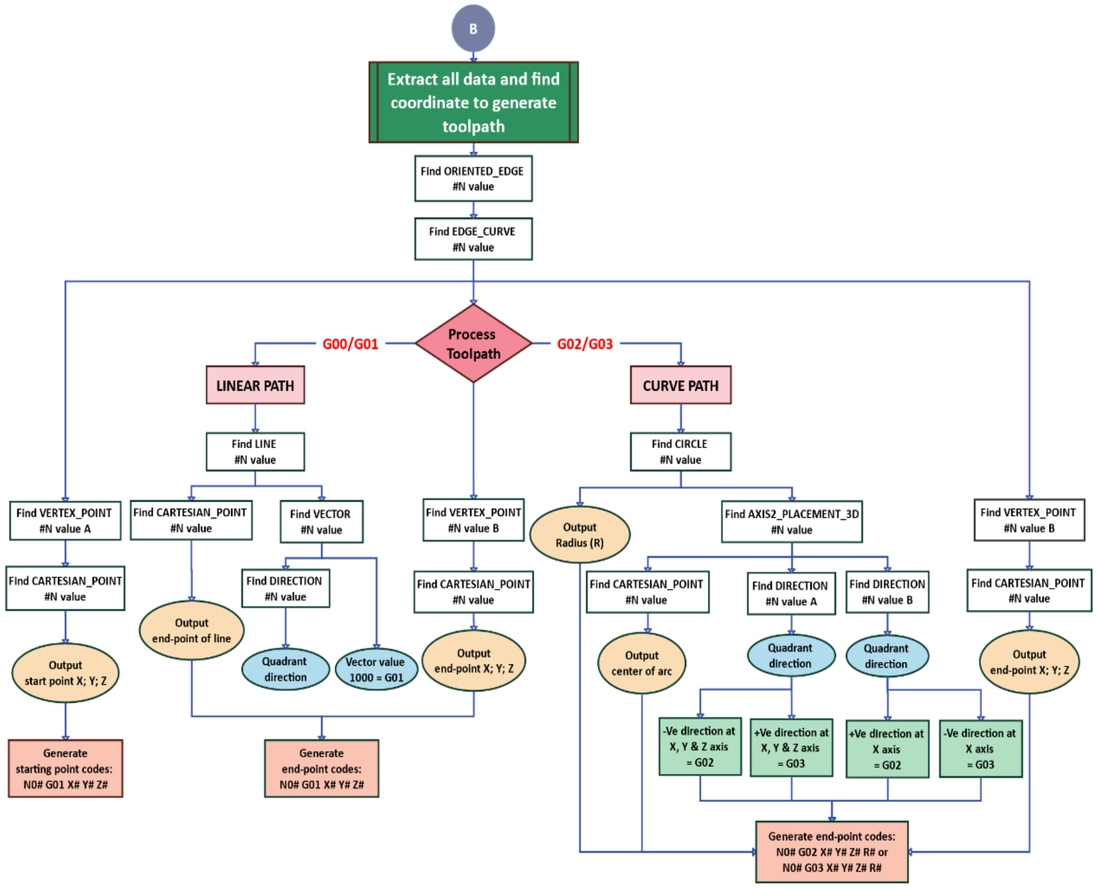

Figure 4. The workflow of profiling path for tool path generation

Analyzing these entities makes it possible to determine the tool path's movement in terms of counterclockwise (CCW), clockwise (CW), or linear motion. Once the movement point of the tool is defined, the reading proceeds to identify four distinct entities with their respective functions: CARTESIAN_POINT, VECTOR, DIRECTION, and AXIS2_PLACEMENT_3D. The data identified through this process is programmed at the end of the reading, and the resulting output follows the international standard arrangement ISO 6983. The determination of G-code plays a critical role in specifying whether the tool path moves counterclockwise (G03), clockwise (G02), or linearly (G01).

3.3. System architecture

The interface development follows a staged approach, focusing on different aspects of program functionality. These stages include interface programming, workpiece size programming, depth of cut programming, edge loop path programming, profiling path programming, looping programming, user input programming, and table of results programming. The interface is implemented using the PHP programming language. The user interface is divided into four sections containing multiple functions and displayed information. The first section has a header function that presents the main titles "INTEGRATED INTERFACE SYSTEM" and "STEP to G-CODE." It also includes the UniSZA and FRIT logos on the left and right sides.

The second section encompasses the core functions of the system. It provides options for uploading STEP files, functions for clearing folders, and processes for clearing databases. Additionally, it displays information such as the workpiece size, minimum X and Y values, depth of cut readings, and a G-code generation function. The third section is dedicated to the "Generate G-Code" function. Activating this function reveals the complete G-code display. This section includes a "Header and Footer" subsection and a "View G-code" process showing the entire program. Users can download and save this G-code program in ".txt" format using the "Download" button. The fourth section in the interface's footer provides information about the I^{2}S program, financial sponsorship, invention, and ownership. This information is presented within the I^{2}S script cluster, as shown in Figure 5.

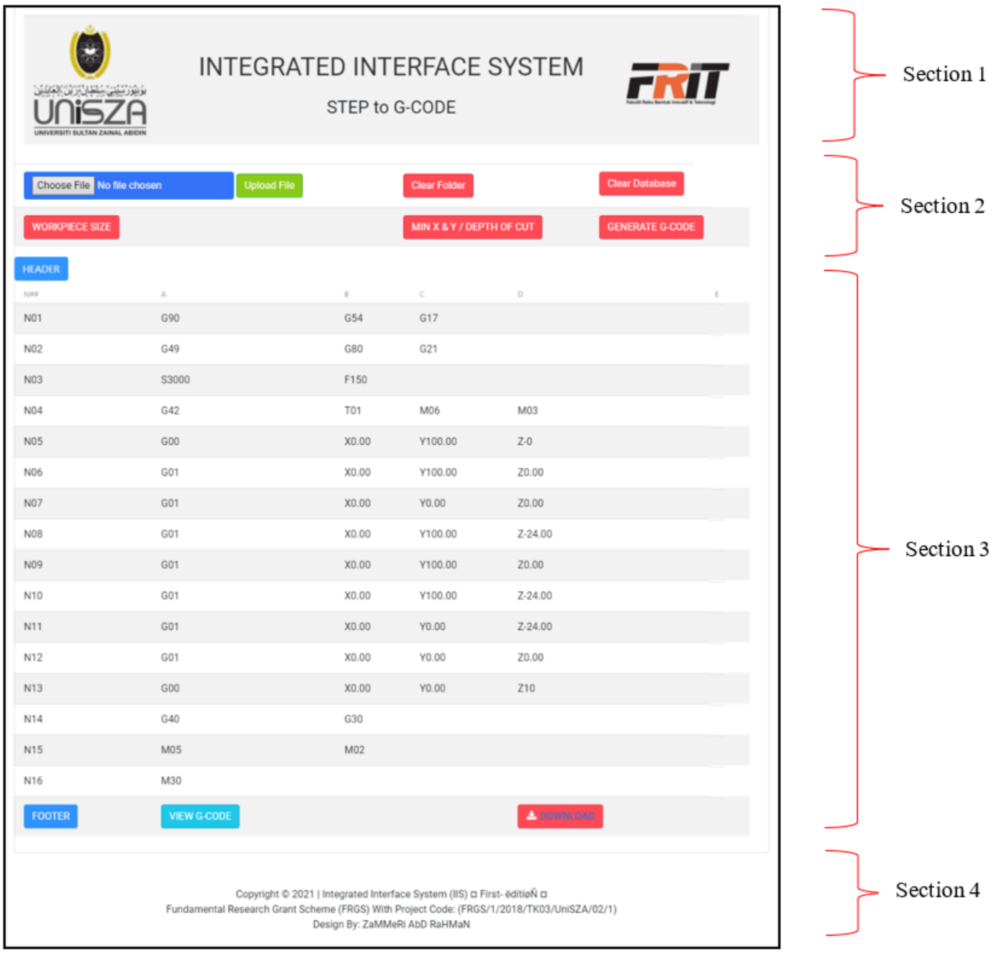

Figure 5. The script cluster in the integrated interface system (I^2S)

4. RESULTS AND DISCUSSION

Validation of the developed workflow, algorithms, and methods for generating G-code programs can be achieved through a case study. By utilizing the readings associated with entity value #599 in the provided sample, the system requirements for tool path movement development can be met. This process enables the assessment of the accuracy of the final product during the machining process, as it relies on precise readings based on actual 3D CAD model drawings. The presence of nineteen values corresponding to various other entities is indicated by the values associated with EDGE_LOOP entity #599 (#305, #1441, #921, #74, #834, #111, #1411, #494, 544, 559, #774, #695, #992, #1140, #1161, #1069, #809, #829, #1299). These values are arranged in their original order and translated into a visual representation, as depicted in Figure 6, and this figure illustrates the sequential positions of the cutting tool according to the provided order.

As illustrated in Figure 4, the workflow algorithm is divided into two main parts, each utilizing specific methods to obtain algorithm readings for straight lines or curved paths. The process of generating tool paths can be further simplified by organizing the steps. Beginning with EDGE_LOOP, categorized as cluster B, all the geometry data from the STEP file can be extracted to determine the coordinate points. For instance, the reading value for entity #305 is obtained by extracting the geometry data from the STEP file to locate the ORIENTED_EDGE entity, followed by identifying the EDGE_CURVE entity. This process leads to the presence of three different entities: VERTEX_POINT, LINE, and CIRCLE. The extracted geometric data for the EDGE_LOOP entity (#305) is represented by the CARTESIAN_POINT, which includes the x, y, and z coordinates. The G-code for lines N06 and N07 is visually presented in Figure 7, showcasing essential elements such as line number "N," G-code instructions, x, y, and z coordinates, and the radius "r." By extracting geometric data in the case study, generating G-code programs using an interface system has become possible. These programs consist of nineteen entities that determine the precise position points for the cutting tool in each line. It is worth emphasizing that the cutter's movement is interpreted counterclockwise, and the cutting depth is set to Z-10.

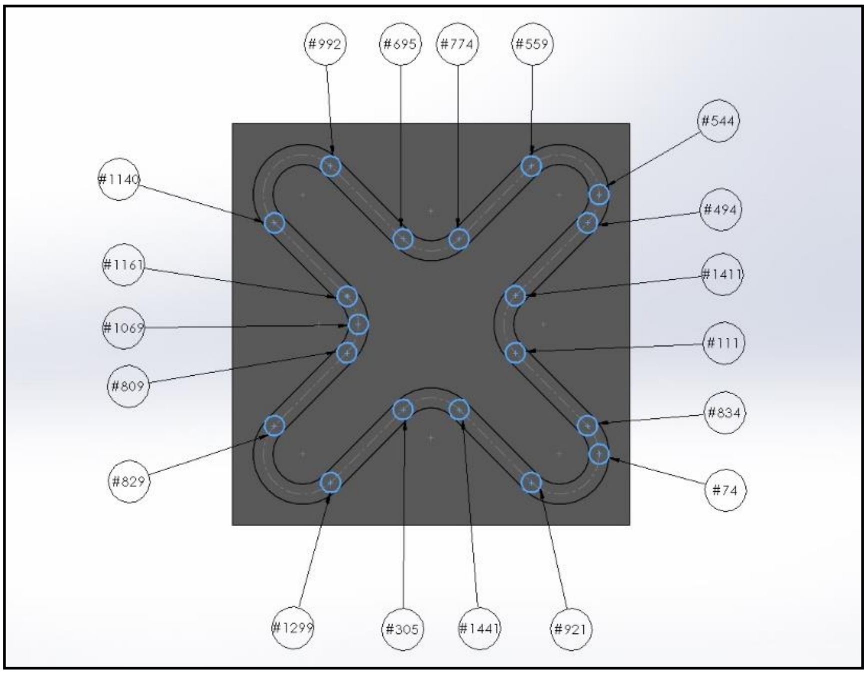

Figure 6. Entity #599 in the EDGE_LOOP array

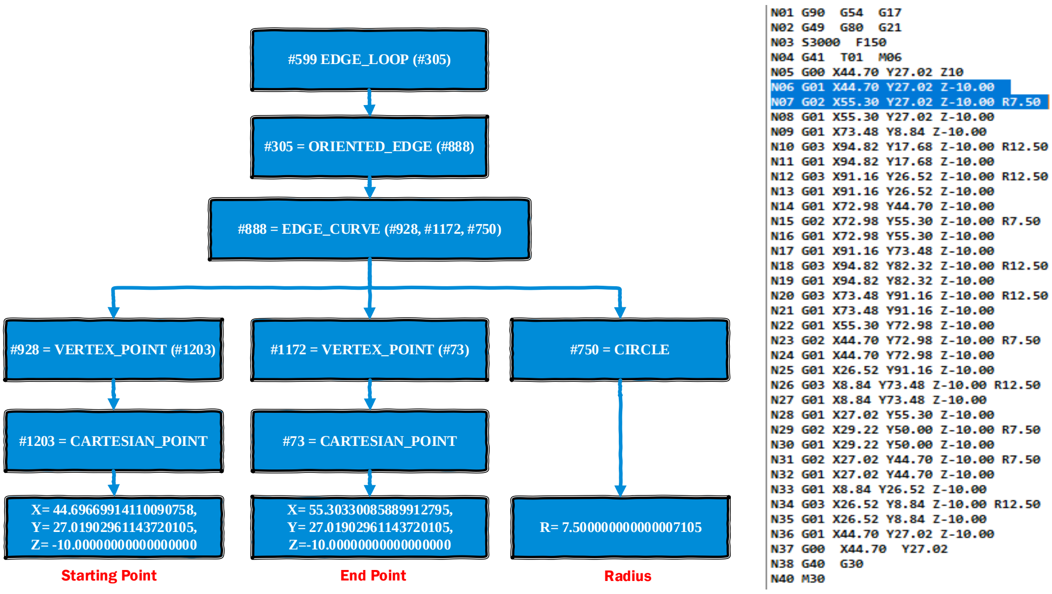

Figure 7. Geometric data extraction diagram for entity EDGE_LOOP (#305)

4.1. Simulation and physical machining result

The results of the simulation process and actual machining for the sample case study are depicted in Figure 8. The image showcases a screen display presenting the movement of the cutting tool and a 3D solid model representation of the object, created using CncSimulator software. After completing the simulation process, physical CNC machining is performed to compare the initial 3D CAD design, simulation results, and the final product. This study sets the spindle speed at 3,000 rpm, and the feed rate is 150 mm/min. The figure demonstrates the machining process of a UHMWPE workpiece utilizing a three-axis CNC milling machine on a benchtop. Ultimately, it can be concluded and verified that the initial CAD drawing, simulation outcome, and finished product all exhibit identical profile shapes in shape and size.

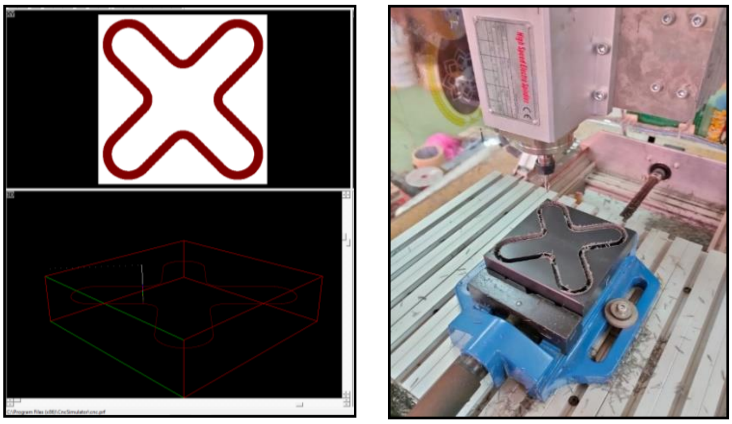

Figure 8. CNC simulation and milling process

5. CONCLUSION

This research aimed to design and create a unified interface system capable of extracting geometric information from STEP files and converting it into G-code for CNC machining operations. The algorithms implemented in this study were written in PHP and executed on a computer. It also introduces a novel technique called “STEP to G-code” to the manufacturing industry. The critical concept in CAD/CAM integration was leveraging STEP files as the driving constraint in CNC programming and utilizing them as input for the machining process. This research proposed an approach that involved comparing results obtained from CNC simulation and physical machining with the profile features of the CAD design. CNC machining and simulation supported the development of the integrated interface system, validating its effectiveness. In addition, the system consistently generated similar outcomes when subjected to testing with CAD models of varying shapes and feature dimensions. It demonstrates the dependability of comprehending the STEP-data structure employed in this study and the construction of the workflow. The created software presents a viable option for converter programming reliant on the ISO 6983 standard, specifically for converting STEP files to G-code. Additionally, it facilitates data integration between the CAD and CAM environments, thereby expediting the product development cycle.

ACKNOWLEDGEMENTS

The authors extend their heartfelt gratitude to University Sultan Zainal Abidin (UniSZA), University Teknikal Malaysia Melaka (UTeM), and University College TATI (UC TATI) for their invaluable support in conducting the research and provision of necessary facilities. Additionally, the authors express their profound appreciation to the Malaysian Ministry of Higher Education (MOHE) for generously funding the project and providing grants through the fundamental research grant scheme (FRGS) (Grant No. FRGS/1/2018/TK03/UNISZA/02/1).

REFERENCES

[1] D. M. Elias, Y. Yusof, and M. Minhat, “CNC machine system via STEP-NC data model and labVIEW platform for milling operation,” in *2013 IEEE Conference on Open Systems, ICOS 2013*, Dec. 2013, pp. 27–31, doi: 10.1109/ICOS.2013.6735042.

[2] Y. Cheng, F. He, X. Lv, and W. Cai, “On the role of generating textual description for design intent communication in feature-based 3D collaborative design,” *Advanced Engineering Informatics*, vol. 39, pp. 331–346, Jan. 2019, doi: 10.1016/j.aei.2019.02.003.

[3] J. Sääski, T. Salonen, and J. Paro, “Integration of CAD , CAM and NC with Step-NC,” 2005. [Online]. Available: http://www.vtt.fi/inf/pdf.

[4] O. E. Ruiz, S. Arroyave, and J. F. Cardona, “EGCL: An extended G-code language with flow control, functions and mnemonic variables,” *International Journal of Computer and Information Engineering*, vol. 6, no. 7, pp. 872–879, 2012, doi: 10.5281/zenodo.1061310.

[5] Y. Liu, G. Zhao, O. Zavalnyi, X. Cao, K. Cheng, and W. Xiao, “STEP-compliant CAD/CNC system for feature-oriented machining,” *Computer-Aided Design and Applications*, vol. 16, no. 2, pp. 358–368, Aug. 2019, doi: 10.14733/cadaps.2019.358-368.

[6] C. Chitsaart, S. Rianmora, M. Rattana-areeyagon, and W. Namjaiprasert, “Automatic generating CNC-code for milling machine,” *International Journal of Industrial and Manufacturing Engineering*, vol. 7, no. 12, pp. 2607–2613, 2013.

[7] M. Minhat, V. Vyatkin, X. Xu, S. Wong, and Z. Al-Bayaa, “A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks,” *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 560–569, Jun. 2009, doi: 10.1016/j.rcim.2008.03.021.

[8] T. K. Nguyen, L. X. Phung, and N. T. Bui, “Novel integration of capp in a g-code generation module using macro programming for CNC application,” *Machines*, vol. 8, no. 4, pp. 1–16, Oct. 2020, doi: 10.3390/machines8040061.

[9] Y. Yusof, N. D. Kassim, and N. Z. Z. Tan, "The development of a new STEP-NC code generator (GEN-MILL)," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 2, pp. 126–134, Feb. 2011, doi: 10.1080/0951192X.2010.531289.
[10] S. T. Newman, R. D. Allen, and R. S. U. Rosso, "CAD/CAM solutions for STEP-compliant CNC manufacture," *International Journal of Computer Integrated Manufacturing*, vol. 16, no. 7–8, pp. 590–597, Jan. 2003, doi: 10.1080/0951192031000115688.
[11] X. W. Xu, L. Wang, and Y. Rong, "STEP-NC and function blocks for interoperable manufacturing," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 3, pp. 297–307, Jul. 2006, doi: 10.1109/TASE.2005.862147.
[12] 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*, vol. 16, no. 3, pp. 1053–1061, Sep. 2022, doi: 10.1007/s12008-021-00837-2.
[13] R. Kirkwood and J. A. Sherwood, "Sustained CAD/CAE integration: integrating with successive versions of step or IGES files," *Engineering with Computers*, vol. 34, no. 1, pp. 1–13, Jan. 2018, doi: 10.1007/s00366-017-0516-z.
[14] L. Jinggao *et al.*, "CAD data exchange using the macro-parametrics approach: an error report," *International Journal of CAD/CAM*, vol. 10, no. 2, pp. 1–6, 2011.
[15] Y. F. Zhao, X. W. Xu, S. Q. Xie, T. R. Kramer, F. M. Proctor, and J. L. Michaloski, "An integrated process planning system architecture for machining and On-Machine Inspection," *Proceedings of the ASME Design Engineering Technical Conference*, vol. 1, no. PARTS A AND B, pp. 589–596, 2008, doi: 10.1115/DETC2008-49725.
[16] H. Ameddah, K. Zidani, and R. Manaa, "Impeller tool paths programming for rough machining in an intelligent NURBS STEP-NC format," *International Journal of Current Engineering and Technology*, vol. 6, no. 1, pp. 194–199, 2016.
[17] X. W. Xu *et al.*, "STEP-compliant NC research: The search for intelligent CAD/CAPP/CAM/CNC integration," *International Journal of Production Research*, vol. 43, no. 17, pp. 3703–3743, Sep. 2005, doi: 10.1080/00207540500137530.
[18] 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, Oct. 2020, doi: 10.30880/ijie.2020.12.07.022.
[19] 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," *The International Journal of Advanced Manufacturing Technology*, vol. 114, no. 9–10, pp. 2549–2566, Jun. 2021, doi: 10.1007/s00170-021-06741-z.
[20] N. Hatem, Y. Yusof, A. Z. A. Kadir, K. Latif, and M. . Mohammed, "A novel integrating between tool path optimization using an ACO algorithm and interpreter for open architecture CNC system," *Expert Systems with Applications*, vol. 178, p. 114988, Sep. 2021, doi: 10.1016/j.eswa.2021.114988.
[21] Z. Abd Rahman, S. B. Mohamed, M. S. Kasim, M. K. A. Ariffin, and A. R. Zulkifli, "The development of methodology for tool path generation using STEP data model based on ISO 6983," *International Journal of Engineering Trends and Technology*, vol. 69, no. 6, pp. 144–153, Jun. 2021, doi: 10.14445/22315381/IJETT-V69I6P222.
[22] 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*, vol. 43, no. 1, 2005, doi: 10.11113/jt.v43.758.
[23] S. B. Mohamed *et al.*, "Integrated interface system for tool path generation of STEP file," in *AIP Conference Proceedings*, 2019, vol. 2129, p. 020012, doi: 10.1063/1.5118020.
[24] J. Ismail, S. B. Mohamed, A. Mohd, and Z. A. 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, doi: 10.55057/ijarti.2021.3.4.6.
[25] Z. A. Rahman *et al.*, "Integrated interface system (I2S) of STEP data models for the CNC milling tool path generation," *International Journal of Integrated Engineering*, vol. 14, no. 4, pp. 160–170, Jun. 2022, doi: 10.30880/ijie.2022.14.04.012.

BIOGRAPHIES OF AUTHORS

Ts. Dr. Zammeri Abd Rahman is currently employed as a senior instructor in the Faculty of Innovative Design and Technology at the Universiti Sultan Zainal Abidin (UniSZA) in Terengganu, Malaysia. He holds a Diploma in Manufacturing Technology from Sultan Zainal Abidin Religions College (KUSZA) and a Bachelor of Manufacturing Engineering (Manufacturing Design) with Honors from the Universiti Teknikal Malaysia Melaka (UTeM). He obtained a Master of Science in Engineering and Engineering Trades, specializing in machine design and analysis at the UniSZA. He then pursued a Ph.D. in the same field and university, focusing on Computer Integrated Manufacturing Systems. His research interests primarily lie in manufacturing technology, CAD/CAM, CNC, STEP files, and machine learning. He can be contacted at email: zammeri@unisza.edu.my.

Professor Ts. Dr. Saiful Bahri Mohamed is a senior lecturer within the Faculty of Innovative Design and Technology at the Universiti Sultan Zainal Abidin (UniSZA) in Terengganu, Malaysia. He holds a Diploma in Industrial Technology from Sultan Zainal Abidin Religions College (KUSZA), a Bachelor of Manufacturing System Engineering from New Castle Upon Tyne Polytechnic in the United Kingdom, and a Master of Computer Integrated Engineering from Loughborough University, also in the United Kingdom. Subsequently, he obtained a Ph.D. in Integrated Manufacturing Systems from the University of Salford. He specializes in various domains, including CAD/CAM, optimization, additive manufacturing, and intelligent systems. He can be contacted at email: saifulbahri@unisza.edu.my.

Associate Professor Ir. Dr. Mohamad Minhat currently holds a senior lecturer position within the Faculty of Manufacturing Engineering at the Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. He completed his Diploma in Industrial Technology at Sultan Zainal Abidin Religions College (KUSZA). Subsequently, he pursued his Bachelor's Honors Engineering degree from the University of Wales College of Cardiff (UWCC) in the United Kingdom and his Master's degree from Coventry University, also in the United Kingdom. He further obtained his Ph.D. from the University of Auckland (UOA) in New Zealand, with a research focus on CAx systems, intelligent, and interoperable manufacturing systems. He can be contacted at email: mohdm@utem.edu.my.

Associate Professor Ts. Dr. Alias Mohd is currently serving as a senior lecturer at the Faculty of Engineering Technology (Manufacturing) at University College TATI (UC TATI) in Terengganu, Malaysia. He holds a Diploma in Industrial Technology from Sultan Zainal Abidin Religions College (KUSZA). He pursued his B. Eng. (Hons.) in Manufacturing Engineering from the University of Sunderland in the United Kingdom. He obtained his M. Eng. in Tooling Engineering from Nettur Technical Training Foundation (NTTF) in Bangalore, Karnataka, India. He earned his Ph.D. in Manufacturing Engineering from Edinburgh Napier University in Edinburgh, Scotland, UK. His areas of specialization encompass tooling, press tool, injection molding, machining, composites polymer, and various manufacturing processes. He can be contacted at email: aliasmohd@uctati.edu.my.

Ts. Zulkifli Abd Rahman currently serves as a senior lecturer at the Faculty of Engineering Technology, University College TATI (UC TATI), Malaysia. He previously held the position of senior lecturer at the Faculty of Engineering and Life Science, Universiti Selangor (UNISEL), Malaysia. He obtained a Bachelor of Engineering in Electrical and Electronics from the Universiti Putra Malaysia (UPM) and a Master of Science in Electronics Engineering from the same institution. Currently, he is pursuing a Ph.D. program at the Universiti Sultan Zainal Abidin (UniSZA), focusing on the convergence of artificial intelligence (AI), robotic process automation (RPA), machine learning (ML), and cognitive platforms. His research interests include embedded controllers, intelligent control systems, artificial intelligence, mobile robots, electrical power, renewable energy, and instrumentation. He can be contacted at email: zulkiflirahman@uctati.edu.my.

Ts. Dr. Roslan Awang is currently employed as a senior lecturer in the Faculty of Innovative Design and Technology at the Universiti Sultan Zainal Abidin (UniSZA) in Terengganu, Malaysia. He holds a B. Eng. (Hons.) in Mechanical Engineering from Universiti Teknologi Malaysia (UTM). He earned his Ph.D. in Technical and Vocational Educational Training (TVET) at Universiti Tun Hussein Onn Malaysia (UTHM). Before migrating to the academic field, he had experience as a production supervisor in several international and local manufacturing industries. Specializing in additive manufacturing, jigs and fixture design, product design development, and various manufacturing processes. He can be contacted at email: roslanawang@unisza.edu.my.