The International Journal of Advanced Manufacturing Technology (2023) 129:5371–5385
https://doi.org/10.1007/s00170-023-12582-9

ORIGINAL ARTICLE

Expert system to implement STEP-NC data interface model on CNC machine

Kamran Latif[1] · **Efrain Rodriguez**[2] · **Renan Bonnard**[3] · **Yusri Yusof**[4] · **Aini Zuhra Abdul Kadir**[5]

Received: 12 March 2023 / Accepted: 22 October 2023 / Published online: 18 November 2023

Abstract

One of the issues in manufacturing is implementing the standard for the exchange of product data-numerical control (STEP-NC) data interface model on computer numeric control (CNC) machines. The most often used STEP-NC programming techniques for this implementation are indirect, interpreted, and adaptive. Because of the ease of integration with existing control systems, the performance of the interpreted method was noticeably superior to that of the indirect and adaptive approaches. This concept has resulted in the creation of several tools, systems, designs, algorithms, and methods. In this study, a new STEP-NC implementation system has been created, in which the interpretation has been done using entity-plus string-based (double layer) for more precise data extractions, the tool paths system can create facing, pocket, drill, bore, ream, countersink, side, slot, and contour operations, the output file generation system can create output as per interpreted and hybrid programming approaches, and the execution system can handle multi-threaded operations. To enhance the overall interpretation system and automate implementation by reducing manual intervention, an expert system has also been incorporated. The STEP-NC part 21 examples 1 and 2 part programs were manufactured on the CNC prototype to validate the technology. The creation of the system, the design of the algorithm, the experimental verification, the conclusion, and the future suggestions are described in the paper's content.

Keywords STEP-NC · ISO 14649 · STEP-NC implementation · CNC expert system

1 Introduction

The CNC unit's ability to generate, parse, and execute sequential control has made it an indispensable part of the production process ever since its inception. The usage of personal computers and CAD/CAM programs is crucial to this innovation. It has seen extensive use across many different sectors, each employing its unique controller and implementation strategy. There was a rise in the 1970s and 1980s in the need for CNC systems to be flexible. Since CNC machines can be easily re-programmed to manufacture a wide variety of components, they have become an invaluable tool for gaining insight into the current state of modular manufacturing [1]. Scalable CNC systems' development did reveal some shortcomings in the ISO 6983 data interface model, such as the following: providing limited data to CNC, shifting one-way communication from CAD/CAM to CNC, being unable to incorporate smooth convergence between CAD-CAM-CNC, having extremely large and difficult to manage programs, and having difficulty accommodating last-minute changes on the shop floor [2]. Over the years, many manufacturers have extended G codes with new supplementary commands to provide new features to their systems. These augmentations are not part of ISO 6983. As a result, G code becomes more machine-specific due to the component programs' effect on computer interoperability [1].

In order to address these concerns, ISO 10303, or the standard for the exchange of product data, was developed in 1994, revised in 2021, and will be replaced with ISO/DIS 10303-1 [3]. This standard aims to offer an open-source methodology for characterizing commodity data at all stages of its existence. The information interchange between CNC machines and CAM systems has been facilitated by ISO 10303, which also increased interoperability between CAD devices. To bring the advantages of STEP to CAM and CNC, the International Organization for Standardization (ISO)

✉ Kamran Latif
kamran@ftv.upsi.edu.my; engr.kamranqureshi@gmail.com

Extended author information available on the last page of the article published ISO 14649 (formerly known as STEP-(numerical control) NC) in 2004 [4]. As a follow-up to ISO 10303, ISO 14649 provides further guidelines regarding CAM and CNC. STEP-NC standard enables the seamless integration of design, engineering, and manufacturing processes by providing a common language for data exchange between various systems and processes. It makes it possible for computer-aided systems (CAx) based on STEP to work in tandem with CNC machines. "Design, manufacture, and support everywhere" is the goal of the standard for the exchange of product data-numerical control (STEP-NC) [1]. Next-generation sophisticated and flexible CNC systems will be possible with the foundation laid by ISO 14649 by offering a standardized data model for the exchange of product and process data in the manufacturing industry. In order to create a smart, connected, and productive manufacturing environment in Industry 4.0, it is essential to incorporate cutting-edge technologies into the manufacturing process, such as the Internet of Things (IoT), artificial intelligence (AI), and data analytics. However, a common language for data exchange between various systems and processes is necessary for this integration. In order to facilitate the seamless exchange of data between various systems and processes, STEP-NC offers this common language. This enables the development of a digital thread connecting every phase of the product lifecycle and serving as a single source of truth for product data and information.

The STEP-NC data model is broken down into two distinct parts: the HEADER and the DATA. The HEADER contains metadata like the file's title, creator, modification date, and owner. In contrast, the DATA part has everything you need to know about production processes and geometrical specifications. The next subsections elaborate on the elements of this section: work plan or executables, technology description, and geometry description. The work plan details a predetermined sequence of manufacturing actions or directions. There could be data on the workpiece in this section as well. Procedure stages, NC functions, and code structures are all examples of executables. The work plan's essential executables define the 2.5D and 3D regions of the manufacturing features. Sub-features, such as pockets, slots, and round holes, with a cutting condition environment are also part of each working stage. The DATA section's technological description includes information on the tool used, the machining technique employed, the workpiece's definitions, the drilled hole's depth, the feed rate, the spindle speed, and the tool's diameter. However, the geometry description is comprehensive, detailing all geometry data employed by the components as per the ISO 10303 format [5, 6]. Figure 1 depicts the overall STEP-NC file structure.

The implementation of STEP-NC faces a number of difficulties, including interoperability, standardization, adoption, complexity, integration with current systems, data management, and cost. Due to STEP-unconventional NC's data structure, integrating it into CNC systems presents a new set of challenges. STEP-NC data interface concept was converted into ISO 6983 to improve its initial performance on CNC machines; this technique is known as the indirect STEP-

Fig. 1 Structure of STEP-NC part 21 physical file

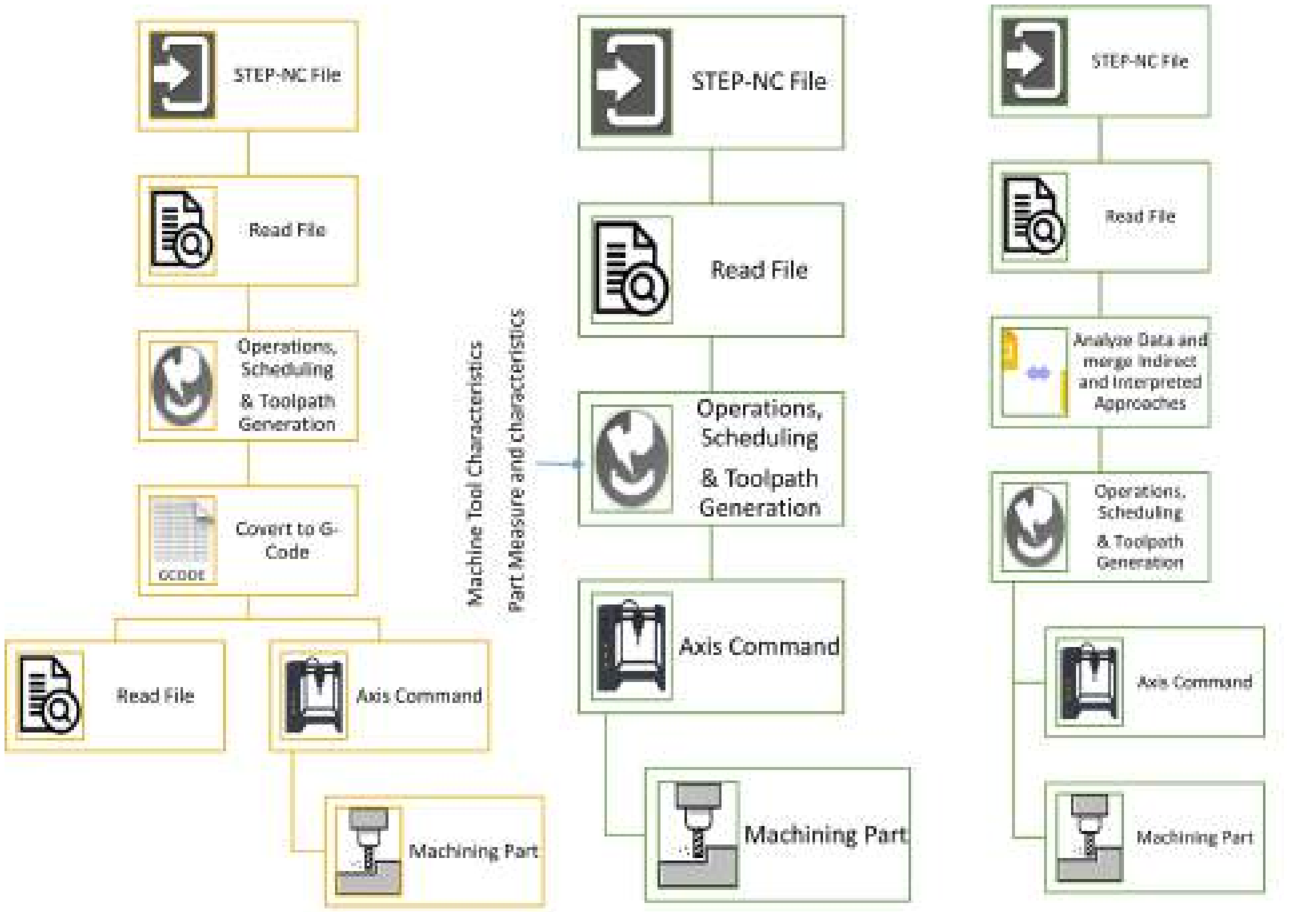
NC programming approach [7]. Multiple iterations of the system have been created using this method, some of which are included in Table 1. These implementations conclude that this translation is insufficient to accomplish the goal of the STEP-NC; hence, a different approach should be employed. The following method relies on interpretation tools to read, analyze, interpret, and generate tool paths from STEP-NC files; this technique is known as the interpreted STEP-NC programming approach [7]. It allowed STEP-NC to be implemented on CNC machines with greater efficiency. Many other systems have been implemented using this method, some of which are included in Table 1, and explained in details with comparison in [8].

Due to its integration with current control systems, the STEP-NC interpretation technique is widely used. Additionally, this connection offers a variety of customization options. In this method, software tools restructured the input data to match the internal hardware structure. Remember that this interpretation is not a G code to STEP-NC conversion. This concept has been put into practice by creating various interpretation tools, systems, designs, algorithms, and procedures. Data extraction, interpretation, and execution on the machine are the main pillars of this field, which are also addressed in this study. The scientific contributions of this study in comparison to [24, 34] are as follows: the interpretation system is enhanced from an entity-based to entity-plus string-based (double layer) for more accurate data extractions; the tool paths system is upgraded to facing, pocket, drill, bore, ream, countersink, side, slot, step, and contour processes; the output file generation system is able to generate interpreted and hybrid programming approaches output file; and the execution system can execute interpreted and hybrid representations. Along with that, the entire interpretation system has been enhanced with expert system to automate and streamline the implementation by reducing the manual intervention.

This research presents a novel method for the STEP-NC implementation system using a virtual component technology-based expert system. The distinctive aspect of this strategy is that it provides two different means of STEP-NC implementation: interpreted and hybrid via expert system. The STEP-NC programming approach described in [7] provides the interpreted method foundation. The hybrid, however, mixes interpreted and indirect STEP-NC

Table 1 Summary of STEP-NC implementation on CNC machine

Indirect STEP-NC systems	Interpreted STEP-NC systems
Aydin et al. [9], 2006, Siemens 840D system for three-axis milling machine developed in JAVA	[10], 2006, PC and PCI7212c system with analog and digital input–output features. Developed in C for three and five-axis control
Lee et al. [10], 2006, PC and PCI7212c system with analog and digital input–output features. Developed in C for three and five-axis control	[11], 2007, PC-based Embedded microcontroller RCM 3700 Linux CNC system developed in C for three and five-axis motion control
Kramer Thomas et al. [12], 2006, FBICS CNC for three-axis milling machine. Developed in C with ST developer and rose library	[13], 2008, PC-based CNC system for three-axis motion control. Developed in PLC, C with ROSE and STIX library
Minhat et al. [14], 2009, Linux PC-based EMC for three-axis motion control. Developed by utilizing JAVA and functional blocks	[15], 2011, OWL and Neural Network-based Linux RTAI system
Sivakumar and Dhanalakshmi [16], 2013, FANUC CNC-based XLTURN controller for the two-axis lathe	[17], 2012, PC and microcontroller-based three-axis CNC prototype developed in JAVA
Minhat et al. [14], Linux CNC for three-axis motion control. Developed in C, STEP-NC modular, and adopter	[18], 2014, HIT-CNC system for three-axis milling machine. Developed in C with CAD/CAM systems
Zhang et al. [19], 2013, Java-based Siemen 840 D controller for three-axis milling machine	[20], 2015, PC, PLC, servo, and Ethernet-based three-axis milling system. Developed with C, PLC, and EtherCAT
Benavente et al. [21], 2013, Siemens Sinumerik 840 Di with Server system for three-axis motion control. Developed in JAVA and ST developer	[22], 2016, an OMAC-based CNC system build with EXPRESS and kernel
Elias et al. [23], 2014, PC-based universal machine interface for three-axis milling machine. Developed in JAVA with ST developer and LabView	[24], 2017, PC-based machine motion control system for three-axis ATC CNC milling machine. Developed in LabVIEW
Saša and Glavonjič [25], 2014, FANUC LOLA with HMC500 machine control system for two and three-axis	[26], 2017, Visual Studio-based Closed-loop STEP-NC system
lvares et al. [27], 2016, Glade JAVA-based Linux CNC for motion control	[28], 2018, ASEA IRB6-S2 based machine control system
Shin et al. [29], 2016, JAVA prime-based Virtual Machining Model (STEP2M) for two-axis motion control	[30], 2019, EXPRESS-based EtherMAC two-axis machine motion control system
Ye et al. [31], 2018, OWL, JAVA, and MapReduce-based PC CNC system	[32], 2021, IEC61499-based STEP-NC motion control system
Liu et al. [33], 2019, Graphtec CNC	



(a) In-Direct STEP-NC Programming Approach (b) Interpreted STEP-NC Programming Approach (c) Hybrid STEP-NC Programming Approach

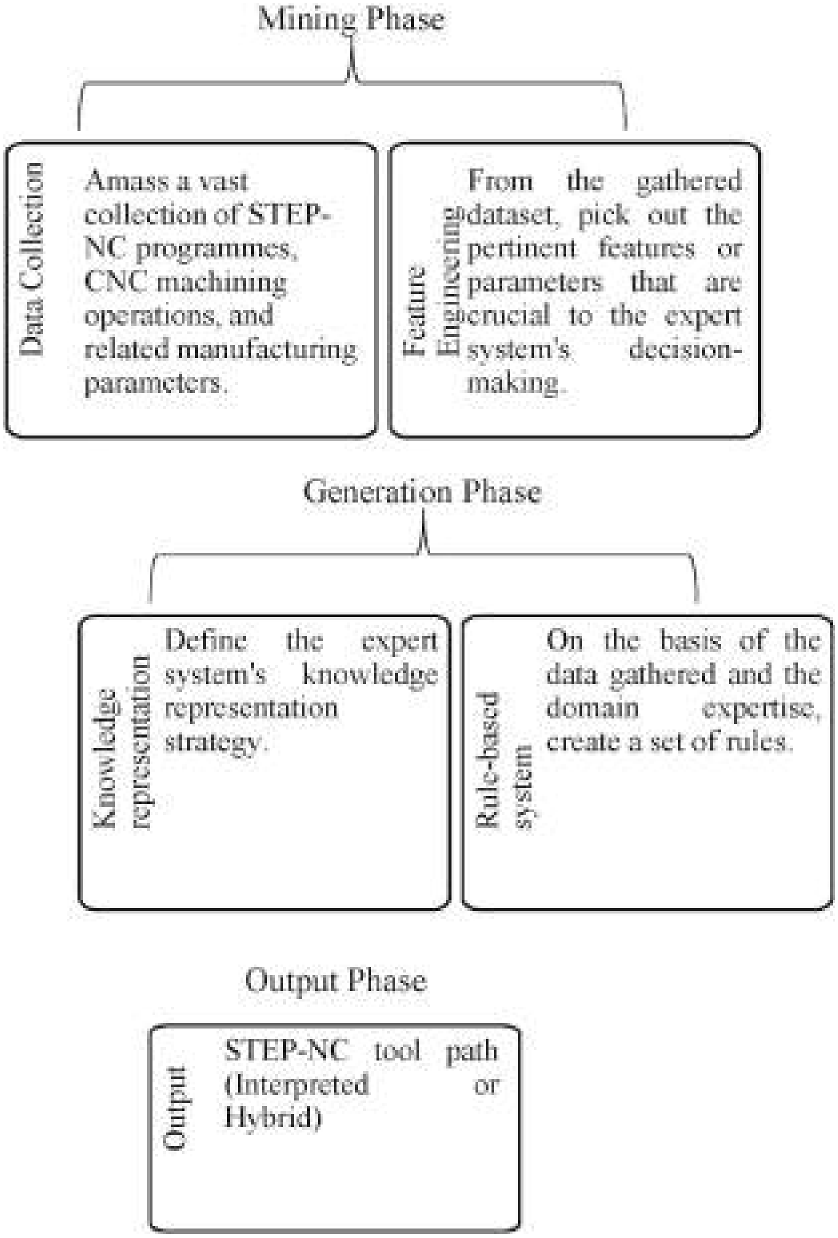
Fig. 2 STEP-NC implementation approaches

programming techniques. Figure 2 provides an overview of all STEP-NC programming techniques. However, an expert system applies STEP-NC by processing and interpreting STEP-NC data using its knowledge base and reasoning abilities. For the purposes of enabling data analysis, process planning, and decision-making, an expert system is trained to comprehend the STEP-NC standard and its application. By considering variables like material properties, tooling, operations, processes, interpretations systems, and machine capabilities, the system uses its knowledge to determine the most effective STEP-NC programming approach to manufacture the part. The further content includes system development, algorithm design, experimental validation, the paper's conclusion, and future recommendations.

2 Expert system

The mining, generating, output, and execution stages make up the expert system. The primary function of an expert system is to enable automated decision-making capabilities in STEP-NC data mining, tool path generation, and machine execution technique. The ISO 14649 data interface model is mined for its useful information in the mining phase so that it can be used in the generation phase during tool path creation. The output phase entails the organization of the created tool pathways into machine-executable code, which is then run on the actual machine in the execution phase. Figure 3 illustrates how the mining, generation, and output phases are used in the expert system's design. The expert system approach begins by collecting information on CNC machining operations, manufacturing parameters, machining scenarios, part geometries, tooling options, machining strategies, and other information. The mined data is analyzed to find the pertinent features or parameters for the decision-making process, such as toolpath details, feed rate, spindle speed, tool specifications, and part geometry. Following feature recognition and data mining, rule-based systems structure the mapping of gathered data and domain knowledge. To create a valid STEP-NC tool path, these rules establish the relationships

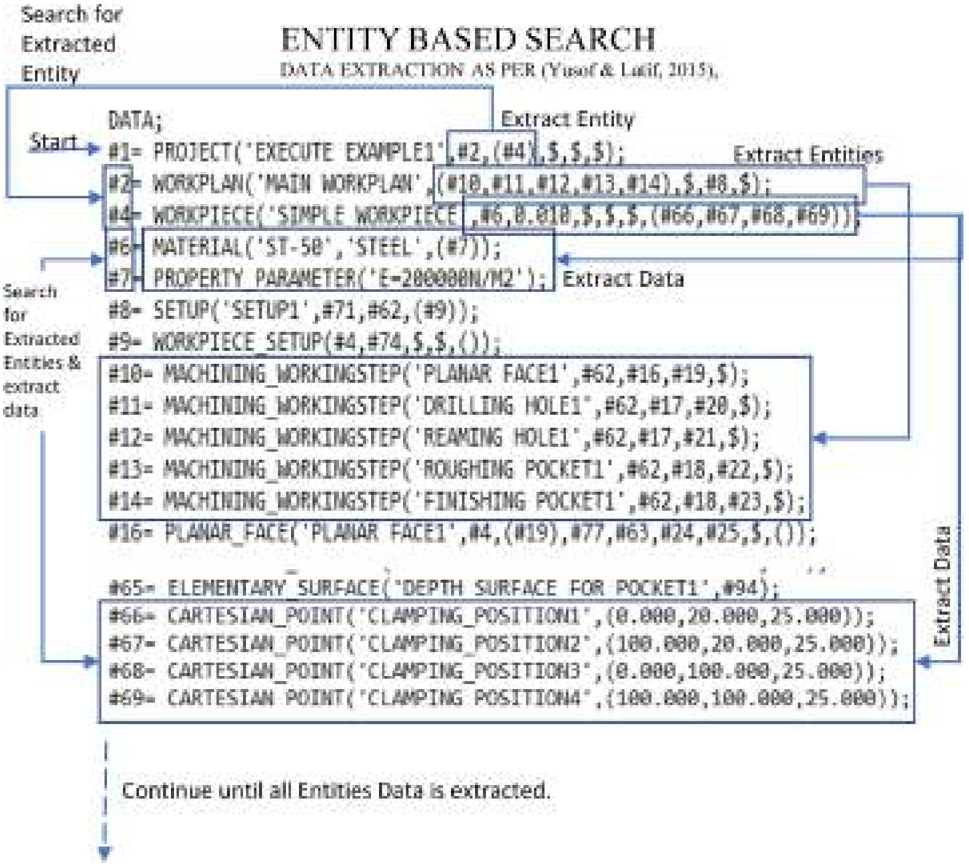
Fig. 3 Expert system design



between the input features and the desired output. The rules are currently defined manually, but in the future, machine learning algorithms will be utilized.

2.1 Mining

The ISO 14649 data interface model code is first parsed by the data mining algorithm, which helps to assure the integrity of the data by ensuring that the data is consistent and error-free and by deleting duplicates. The mined data is used to train the expert system. The entire data extraction process is carried out in two layers, namely, entity-based extraction and string-based extraction. In the entity-based data extraction, a STEP-NC file is broken down into its component entities, each entity represents a different aspect of the product model data, such as geometric shapes, machining operations, toolpaths, and material properties. The data extraction process typically involves parsing the STEP-NC file, identifying the relevant entities, and extracting the data from each entity. Whereas in the string-based data extraction, the process of extracting data from a STEP-NC file is carried out by reading and parsing the string-based representation of the data in the file. In a STEP-NC file, data is stored in an ASCII text format, with each line of the file representing a different aspect of the product model data. The data extraction process typically involves reading the string-based data from the file, extracting the data into its component elements.



```
DATA;
#1= PROJECT('EXECUTE EXAMPLE1',#2,(#4),$,$,$);
#2= WORKPLAN('MAIN WORKPLAN',(#10,#11,#12,#13,#14),$,#8,$);
#4= WORKPIECE('SIMPLE WORKPIECE',#6,0.010,$,$,$,(#66,#67,#68,#69));
#6= MATERIAL('ST-50','STEEL',(#7));
#7= PROPERTY_PARAMETER('E=200000N/M2');
#8= SETUP('SETUP1',#71,#62,(#9));
#9= WORKPIECE_SETUP(#4,#74,$,$,());
#10= MACHINING_WORKINGSTEP('PLANAR FACE1',#62,#16,#19,$);
#11= MACHINING_WORKINGSTEP('DRILLING HOLE1",#62,#17,#20,$);
#12= MACHINING_WORKINGSTEP('REAMING HOLE1',#62,#17,#21,$);
#13= MACHINING_WORKINGSTEP('ROUGHING POCKET1',#62,#18,#22,$);
#14= MACHINING_WORKINGSTEP('FINISHING POCKET1',#62,#18,#23,$);
#16= PLANAR_FACE('PLANAR FACE1',#4,(#19),#77,#63,#24,#25,$,());

#65= ELEMENTARY_SURFACE('DEPTH SURFACE FOR POCKET1',#94);
#66= CARTESIAN_POINT('CLAMPING_POSITION1",(0.000,20.000,25.000));
#67= CARTESIAN_POINT('CLAMPING_POSITION2",(100.000,20.000,25.000));
#68= CARTESIAN_POINT('CLAMPING_POSITION3",(0.000,100.000,25.000));
#69= CARTESIAN_POINT('CLAMPING_POSITION4",(100.000,100.000,25.000));
```

Fig. 4 Entity-based data extraction

The entity-based approach is designed to begin by looking for entity number (#1), which serves as the project data's entry point. The algorithm will take the entity number's (#1) whole data and separate it into entity numbers and entity data tokens. When all of the data has been extracted, as illustrated in Fig. 4, the extracted entities will be searched once again to yield further data. The entity-based technique's specifics are described in [34], whereas the string-based is explained here. The input STEP-NC code's general information can be found in the header section, while geometry, topology, tooling, and machining information can be found in the data section. Therefore, the algorithm divides the header and data sections into separate files. From them, the algorithm sorts the information about the workpiece and the work plan entities string separately and then also looks for the project entity string within the data section. Information about the material, material properties, and clamping positions is then extracted from the workpiece entity string. The working steps data is accessed by removing the work plan entity string, which is then split into the entity strings for the security plane, the cutting process, and the finishing process. Data on the security plane's elementary surface, axis placement, location, and reference direction can be gleaned from its entity string by further extracting it. The cutting process entity string extraction gleans data on the workpiece, finishing process, axis location, depth, axis, and the reference direction. Last but not least, information about the cutting tool, machining technology, machine function, and machining strategy is transferred from the extraction of the finishing process entity string. After all relevant information has been extracted, the algorithm then generates a data cluster for each machining process. Then, the data clusters of entity-based and string-based are matched together to authenticate the data. In the case of complete data match, the cluster data is merged as a single data and parsed to train the expert system, and later to be used in the subsequent generation of tool paths. Figure 5 shows the overall structure of the algorithm design used in this stage. Figures 4 and 5 make the difference between these two methods very clear. The ISO 14649 data contains entity strings (workpiece, work plan, etc.) as well as entity numbers (#13, #14 etc.). The entity-based data extraction method uses entity numbers to find and extract the data. In contrast, the string-based approach extracts data by adhering to entity strings.

2.2 Generation

The cluster information is used by the generation algorithm to produce tool paths that can be executed automatically. This section introduces a few of the many algorithms used to generate tool paths. The developed expert system

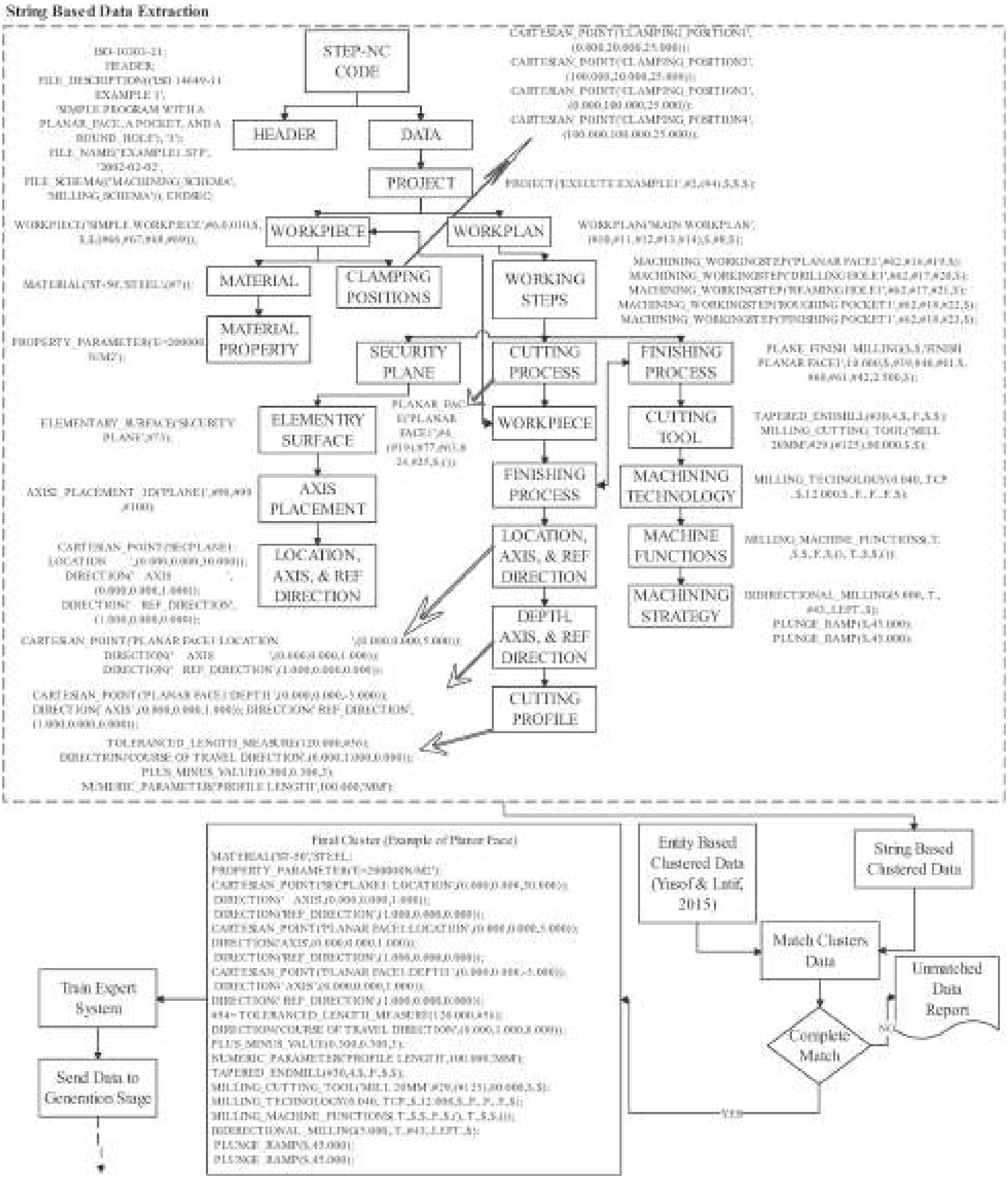


Fig. 5 String-based data extraction and algorithm design of the data mining phase

first decomposes the cluster information into its constituent parts, such as geometry, location, depth, profile, tool, and machining classes, before it can be processed into tool path algorithms. Once the tool paths are created, the expert system optimizes the machining time based on the concept presented in [35].

2.2.1 Facing and pocket operation

The cluster provides the algorithm with information on the facing and pocket operations, including location, depth, profile, and machining data. Based on this information, the expert system determines the optimal tool position (inside, center, outside) and cutting strategy for the geometry. The expert system can generate the existing facing and pocket tool path in three ways: zigzag, one-way, and contour. The algorithm will begin constructing tool paths based on cluster information after these selections and perform optimization at the end. The jargon used in this procedure is depicted in Fig. 6.

2.2.2 Drill, bore, counter sink, and ream

Data on the location, depth, profile, and machining of drills, bores, reamers, and counter sinks are retrieved from the cluster and compared with data on the corresponding holes to determine the algorithm's tool path strategy. The system will use a circle mill strategy if the tool diameter is smaller than the hole diameter. However, when basic drilling is appropriate, it is used. Figure 7 shows the developed system design and working mechanism.

2.2.3 Side, slot, step, and profile contour

The algorithm constructs the tool paths by retrieving the location, depth, profile, and machining data from the cluster and applying it to the side, slot, step, and contour results. In order to guarantee complete material removal, the system optimizes geometrical and profile data to determine the proper amount of overlap in the process profiles. Figure 8

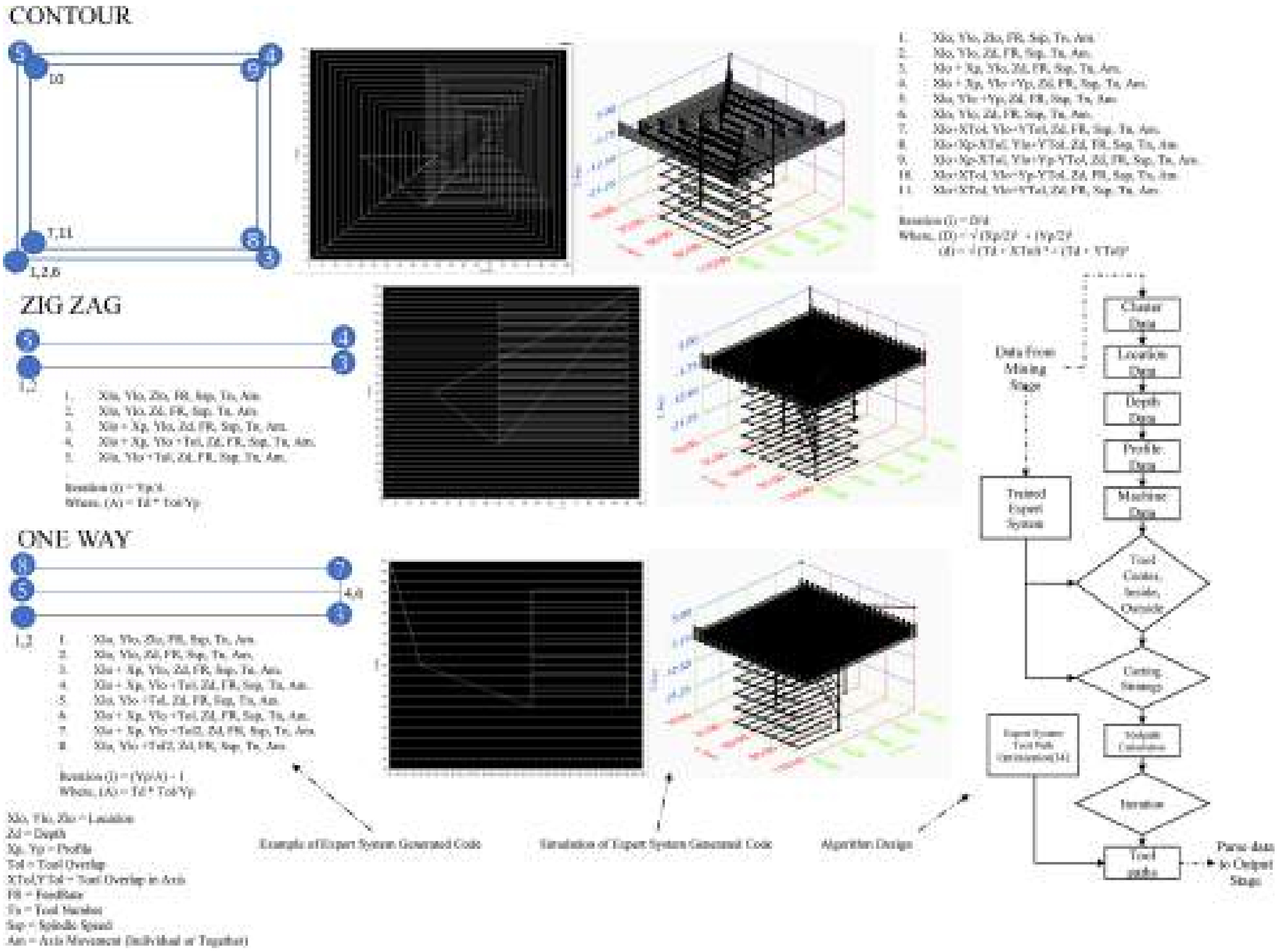


Fig. 6 Facing and pocket generation

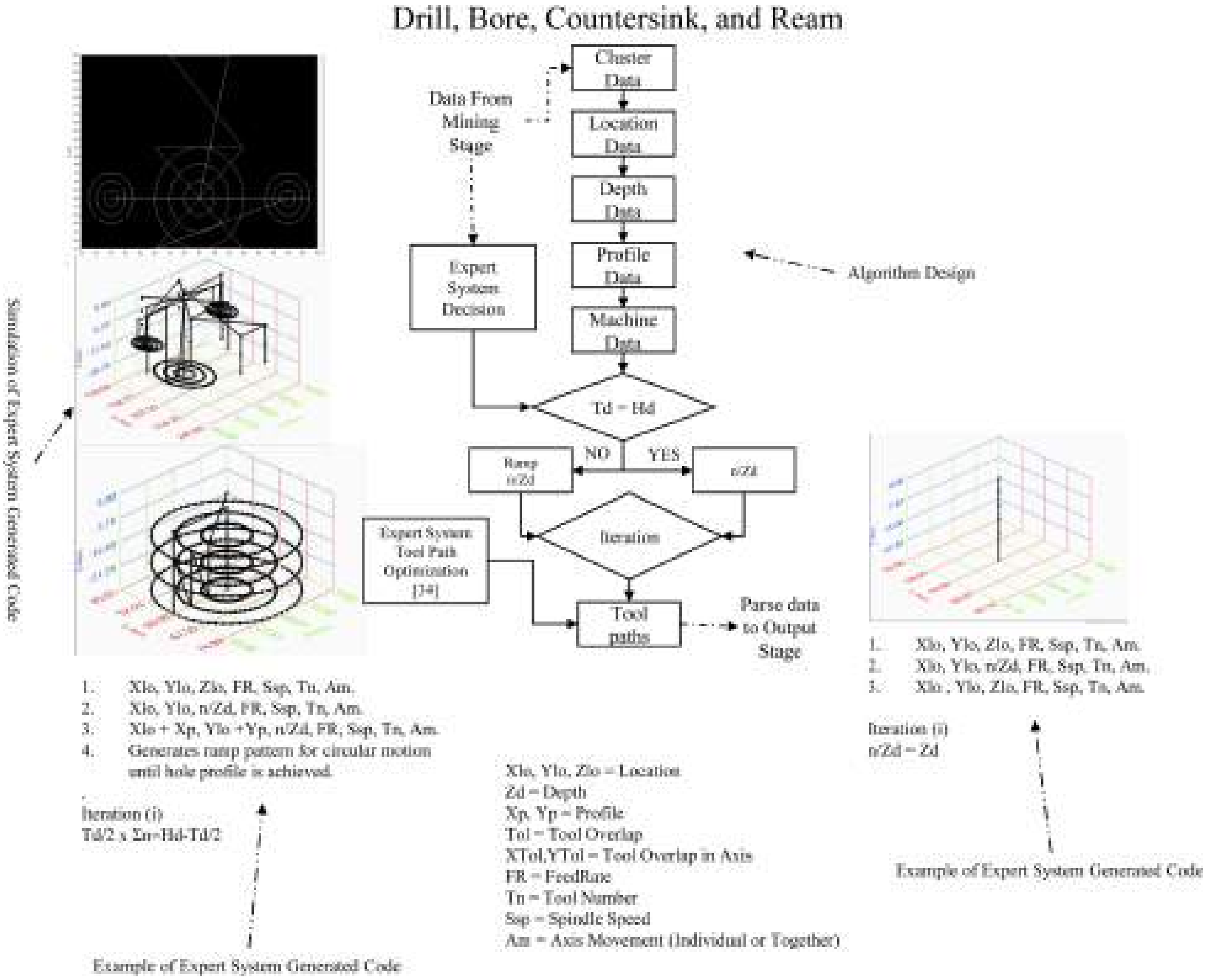


Fig. 7 Drill, bore, counter sink, and ream generation

depicts the framework used by the algorithms underlying these procedures.

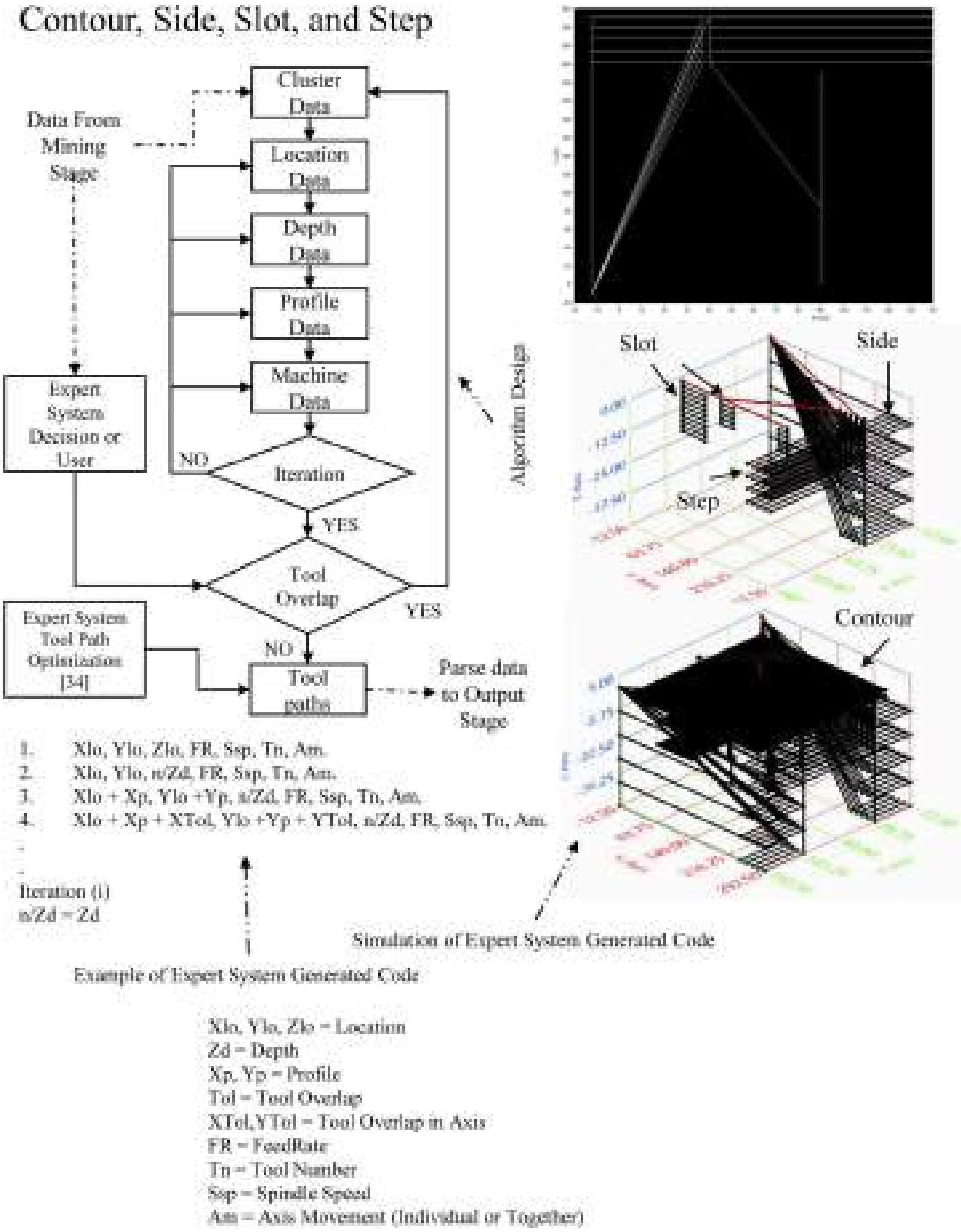
2.3 Output

This stage follows tool path generation and converts the resulting data into code that can be run on a CNC machine in accordance with ISO standards. The two distinct machine-executable code types that the expert system can generate are interpreted and hybrid. The information in the interpreted code is in the special format required to run the machines using coordinates. This translation is illustrated by the codes depicted in the tool paths in Figs. 5, 6, and 7. The developed CNC prototype used it to carry out actual machine execution; this is discussed in greater detail in the experimental study section. Indirect and interpreted forms come together to form a hybrid code. When using just indirect or interpreted instructions is not enough to complete the task, the expert system creates a hybrid code to do it. The output code is generated in a mix of indirect and interpreted formats specified by the expert system.

2.4 Execution

During this stage, the algorithm examines the code in its machine-readable form. The process of deciding which method to use in implementation will begin once it has been determined that the code can be run on the target machine. Algorithms use an interpreted approach-based machine execution system if their output code is written in that style. The algorithm will similarly decide on indirect or hybrid machine execution strategies. The hybrid strategy is utilized by the algorithm if certain functionality is outside the scope of the interpreted approach which will combine

Fig. 8 Side, slot, step, and profile contour generation



the indirect and interpreted methods for tool path building. After this decision is made, the information pertaining to the various cuts is separated so that the cutting process can be completed without interference from other cuts. A program stores the data and transforms it into blocks, which are then used as input by the communication and feedback program. These blocks contain the entire manual for building the machine. The communication tool initiates the connection between the execution system and machine hardware to perform motion control operations. The appropriate parts of the machine are informed of things like axis motion, cutting tool, spindle, and feed. Meanwhile, the data is sent back to the executable application for tracking purposes. The editor tool receives the machine-executable data and updates the shop floor while the machine is operating. Through the use of a human-machine interface (HMI), the device can be run automatically by a computer and manually by a human operator. The visual simulator and machine monitoring tools are activated when the code is run, providing live streaming of the cutting process alongside sensor data. The execution system only feedbacks the machine-executable data for shop flour editing, whereas the sensor data is used for live monitoring only. This feedback of information is limited to execution system; it is not feedback to expert system. The flowchart for the algorithm used in the execution is shown in Fig. 9.

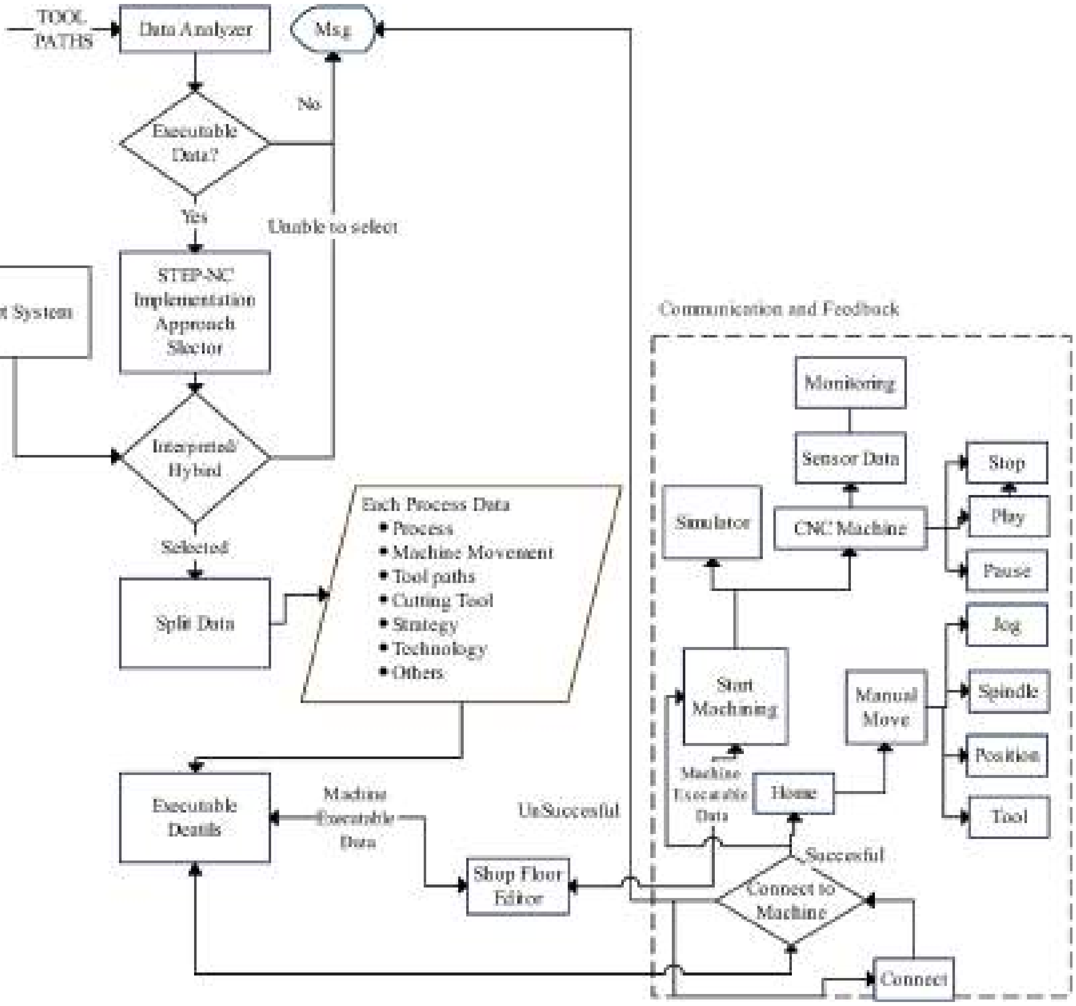


Fig. 9 Execution system algorithm design

3 System interface

As can be seen in Fig. 10, the system has two primary user interfaces: generation and execution. There are tabs labeled “interpreter,” “simulator,” and “toolpath” in the generation interface. Data mining, tool path generation, machine-executable code, and code modifications are the main features of the interpreter tab. Comparatively, the simulator tab edits the generated code and runs graphical simulations in 2D and 3D. Detailed information about the cutting process is available in the toolpath tabs and can be used for in-shop tweaking. Jog, start, stop, pause, single step, home, reset zero, return to zero, soft reset, configuration, clear alarm, spindle, video stream, and capture are some features available through the execution interface. The tool path data is automatically uploaded into the execution system, and the cutting process is initiated upon pressing the start button because the generated code already contains all the necessary information about the machine setup and cutting process. All control buttons can also be used to manually operate the machine instead of having it read commands from the input code.

4 Experimental validation

The developed system was put to the test on a STEP CNC prototype using the Part 21 examples 1 and 2 programs. The prototype consists of a motion control card, microcontrollers, a stepper motor, a spindle, motor drivers, and a custom circuit board. The code for the motion control boards is based on virtual component technology and the concept of execution algorithm. Face, pocket, and drill procedures are all present in the first test file. Cutting operations such as facing, drilling, boring, tapping, pocket, slot, t-slot, and step are all represented in the second STEP-NC sample file. The example file was executed in two case studies based on the interpreted and hybrid approaches. In the interpreted approach, all

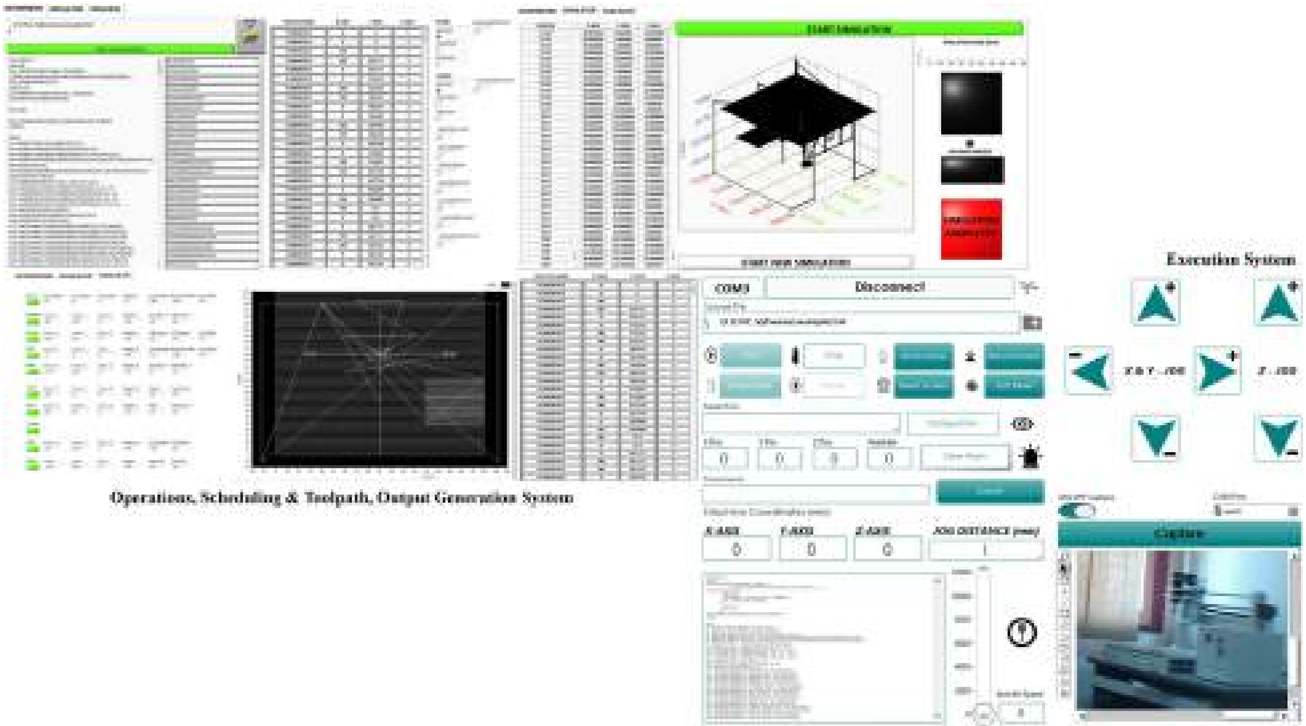
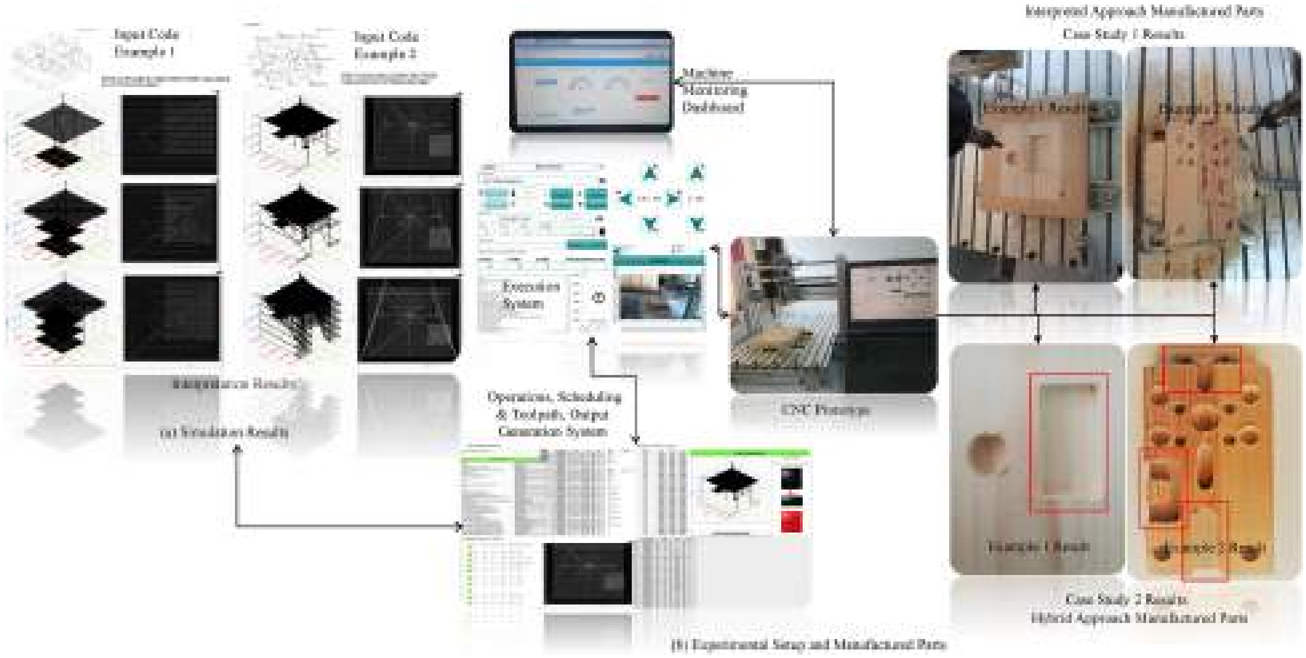


Fig. 10 System interface

manufacturing operations were directly executed by the execution system, whereas, to test the hybrid system pocket, slot, side, and contour operations were disabled from the system database so the expert could generate the hybrid code. The simulation outcomes for the two cases discussed above are shown in Fig. 11a. The tool path generation by the expert system was found to be satisfactory as the manufactured parts are the same in design and dimension in both cases. Figure 11b shows the manufactured parts and the entire experimental process. The successful manufacturing of the parts validates



(a) Simulation Results

(b) Experimental Setup and Manufactured Parts

Fig. 11 Experimental process and results the developed expert system capabilities of mining, generation, output, and execution.

5 Conclusion

Since its inception, STEP-NC has caused functional difficulties for CNC machine builders. Indirect, interpreted, and adaptive methods, among others, have been utilized to carry out this application. In order to execute the machine via the indirect method, the STEP-NC file must be translated back to G code. At the same time, interpreted methods produce the tool path based on the STEP-NC data they are given. This strategy relied on numerous pieces of software and computational processes to carry out the interpretation process. These two implementations of STEP-NC, which make use of a wide range of software and hardware, have been the basis for the creation of a wide variety of systems over the period of several decades. The interpretive method consistently outperformed the indirect method for the duration of the project's execution. But there needs to be a lot of research done in this area. The work done in this research improved the STEP-NC programming method system by creating new algorithms and systems. New algorithms for data mining, tool path development, and execution are added to the indirect and interpreted STEP-NC programming paradigm. If certain functionality is outside the scope of the present system, a new hybrid method is presented, which will combine the indirect and interpreted methods for tool path building. An alternative method for developing machine-executable code in a bespoke format is also introduced. A two-tiered approach to data extraction has also been implemented for further improvement. Furthermore, a new STEP-NC execution system based on virtual component technology is developed. The execution system may run the STEP-NC component program in both interpreted and hybrid modes. There is a whole new algorithm involved in this way of implementation design, and it can be run on machines in real time. Tools for modeling and monitoring, as well as remote manipulation, are a part of the package. The two implemented systems were subjected to experimental testing, with examples 1 and 2 from STEP-NC part 21 being executed. In the interpreted approach, all manufacturing operations were directly executed by the execution system, whereas, to test the hybrid system pocket, slot, side, and contour operations were disabled from the system database so the expert could generate the hybrid code. The successful manufacturing of the parts validates the developed expert system capabilities of mining, generation, output, and execution. In the near future, an adaptive STEP-NC implementation strategy will be incorporated into the system, along with more optimization and artificial intelligence algorithms to enhance system capabilities in light of the fourth industrial revolution.

Acknowledgements The authors wish to thank the Malaysian Ministry of Higher Education for supporting this project under the Fundamental Research Grant Scheme (FRGS). The authors also acknowledge the contributions of the Department of Engineering Technology, Fakulti Teknikal Dan Vokasional, Universiti Pendidikan Sultan Idris (UPSI), Department of Mechanical Engineering University of Brasilia, National Service for Industrial Apprenticeship SENAI, Senai Innovation Institute For Embedded Systems, Universiti TeknikalMalaysiaMelaka (UTeM), the Advanced Manufacturing and Material Centre(AMMC) of Universiti Tun Hussein Onn Malaysia (UTHM), and the Production Laboratory of Universiti Teknologi Malaysia (UTM).

Availability of data and materials All data generated or analyzed during this study are included in this published article and available at the corresponding author.

Code availability Custom codes are available at the corresponding author.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Xu XW, Newman ST (2006) Making CNC machine tools more open, interoperable and intelligent-a review of the technologies. Comput Ind 57(2):141–152. https://doi.org/10.1016/j.compind.2005.06.002
2. Suh S-H, Cheon S-U (2002) A framework for an intelligent CNC and data model. Int J Advan Manuf Technol 19:727–735. https://doi.org/10.1007/s001700200083
3. ISO 10303-1. Industrial automation systems and integration-product data representation and exchange-part 1: overview and fundamental principles. International Standard, ISO/TC 184/SC 4, (1994 and 2021). https://www.iso.org/standard/72237.html
4. ISO (International Standards Organization) (2004) ISO 14649: industrial automation systems and integration-physical device control-data model for computerized numerical controllers, Part 10: General Process Data. https://www.iso.org/standard/40895.html
5. Ridwan F, Xun X, Liu G (2012) A framework for machining optimisation based on STEP-NC. J Intell Manuf 23:423–441. https://doi.org/10.1007/s10845-010-0380-9
6. Latif K, Yusof Y, Kadir AZA (2022) Development of virtual component-based STEP-compliant CNC system. Prog Add Manuf 7(1):77–85. https://doi.org/10.1007/s40964-021-00215-0
7. Rauch M, Laguionie R, Hascoet J-Y, Suh S-H (2012) An advanced STEP-NC controller for intelligent machining processes. Robot Comput-Integr Manuf 28(3):375–384. https://doi.org/10.1016/j.rcim.2011.11.001
8. Latif K, Adam A, Yusof Y, Kadir AZA (2021) A review of G code, STEP, STEP-NC, and open architecture control technologies based embedded CNC systems. Int J Adv Manuf Technol 114:2549–2566. https://doi.org/10.1007/s00170-021-06741-z
9. Aydin N, Newman ST, Allen RD (2006) The application of multi-agent systems for STEP-NC computer aided process planning of prismatic components. Int J Mach Tool Manuf 46(5):559–574. https://doi.org/10.1016/j.ijmachtools.2005.06.005
10. Lee W, Bang Y-B, Ryou MS, Kwon WH, Jee HS (2006) Development of a PC-based milling machine operated by STEP-NC in

XML format. Int J Comput Integr Manuf 19(6):593–602. https://doi.org/10.1080/09511920600623674

11. Francesco C, Celentano G (2007) Design and realization of a STEP-NC compliant CNC embedded controller. IEEE conference on emerging technologies and factory automation, pp 1010–1017. https://doi.org/10.1109/EFTA.2007.4416894
12. Kramer Thomas R, Proctor F, Xun Xu, Michaloski JL (2006) Run-time interpretation of STEP-NC: implementation and performance. Int J Comput Integr Manuf 19(6):495–507. https://doi.org/10.1080/09511920600622056
13. Lan H, Liu R, Zhang C (2008) A multi-agent-based intelligent STEP-NC controller for CNC machine tools. Int J Prod Res 46(14):3887–3907. https://doi.org/10.1080/00207540701213494
14. Minhat M, Vyatkin V, Xun X, Wong S, Al-Bayaa Z (2009) A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks. Robot Comput-Integr Manuf 25(3):560–569. https://doi.org/10.1016/j.rcim.2008.03.021
15. Li P, Tianliang H, Zhang C (2011) A unified communication framework for intelligent integrated CNC on the shop floor. Procedia Eng 15:840–847. https://doi.org/10.1016/j.proeng.2011.08.156
16. Sivakumar S, Dhanalakshmi V (2013) A feature-based system for CAD/CAM integration through STEP file for cylindrical parts. Indian J Eng Mater Sci (IJEMS) 20:21–26. http://nopr.niscpr.res.in/handle/123456789/16151
17. Pacheco NDO, Eduardo H, Rosso RS Jr, Hounsell MDS, Ferreira JCE (2012) Application of the step-NC standard in a computer numerical controlled machining device. In: ABCM symposium series in mechatronics, vol 5, pp 713–723. https://abcm.org.br/symposium-series/SSM_Vol5/Section_IV_Industrial_Informatics_Discrete_and_Hybrid_Systems/05322.pdf
18. Hu P, Hongya F, Zhenyu H, Dedong H (2014) A closed-loop and self-learning STEP-NC machining system. IEEE/ASME international conference on advanced intelligent mechatronics, pp 1598–1603. https://doi.org/10.1109/AIM.2014.6878312
19. Zhang X, Nassehi A, Safaieh M, Newman ST (2013) Process comprehension for shopfloor manufacturing knowledge reuse. Int J Prod Res 51(23–24):7405–7419. https://doi.org/10.1080/00207543.2012.757669
20. Xiao W, Zheng L, Huan J, Lei P (2015) A complete CAD/CAM/CNC solution for STEP-compliant manufacturing. Robot Comput-Integr Manuf 31:1–10. https://doi.org/10.1016/j.rcim.2014.06.003
21. Benavente JC, Ticona JC, Ferreira E, Goulart CM, de Oliveira VG (2013) A STEP-NC compliant system for the remote design and manufacture of mechanical components through the Internet. Int J Comput Integr Manuf 26(5):412–428. https://doi.org/10.1080/0951192X.2012.719086
22. Po H, Han Z, Fu Y, Fu H (2016) Implementation of real-time machining process control based on fuzzy logic in a new STEP-NC compatible system. Math Prob Eng. https://doi.org/10.1155/2016/9814973
23. Elias DM, Yusof Y, Minhat M (2014) An open STEP-NC controller via LabVIEW platform. In: Applied mechanics and materials, vol 660, pp 873–877, Trans Tech Publications Ltd. https://doi.org/10.4028/www.scientific.net/AMM.660.873
24. Latif K, Yusof Y, Nassehi A, Latif QBAI (2017) Development of a feature-based open soft-CNC system. Int J Adv Manuf Technol 89:1013–1024. https://doi.org/10.1007/s00170-016-9124-0
25. Saša Ž, Glavonjič M (2014) Methodology for implementation scenarios for applying protocol STEP-NC. J Prod Eng 17(1):71–74. https://machinery.mas.bg.ac.rs/handle/123456789/4974
26. Lei P, Zheng L, Xiao W, Li C, Wang D (2017) A closed-loop machining system for assembly interfaces of large-scale component based on extended STEP-NC. Int J Adv Manuf Technol 91:2499–2525. https://doi.org/10.1007/s00170-016-9904-6
27. lvares Á, José A, Paredes MEG, Ferreira JCE, Benavente JCT, (2016) A web-based STEP-NC-compliant architecture for low cost 3D part manufacturing. Int J Manuf Res 11(1):1–27. https://doi.org/10.1504/IJMR.2016.076975
28. Toquica JS, Živanovič S, Alvares AJ, Bonnard R (2018) A STEP-NC compliant robotic machining platform for advanced manufacturing. Int J Adv Manuf Technol 95:3839–3854. https://doi.org/10.1007/s00170-017-1466-8
29. Shin S-J, Woo J, Kim DB, Kumaraguru S, Rachuri S (2016) Developing a virtual machining model to generate MTConnect machine-monitoring data from STEP-NC. Int J Prod Res 54(15):4487–4505. https://doi.org/10.1080/00207543.2015.1064182
30. Gao W, Zhang C, Tianliang H, Ye Y (2019) An intelligent CNC controller using cloud knowledge base. Int J Adv Manuf Technol 102:213–223. https://doi.org/10.1007/s00170-018-03222-8
31. Ye Y, Tianliang H, Zhang C, Luo W (2018) Design and development of a CNC machining process knowledge base using cloud technology. Int J Adv Manuf Technol 94:3413–3425. https://doi.org/10.1007/s00170-016-9338-1
32. Harbs E, Negri GH, Jarentchuk G, Hasegawa AY, Rosso RSU Jr, da Hounsell SM, Lafratta FH, Ferreira JC (2021) CNC-C2: an ISO14649 and IEC61499 compliant controller. Int J Comput Integr Manuf 34(6):621–640. https://doi.org/10.1080/0951192X.2021.1911002
33. Liu Y, Zhao G, Zavalnyi O, Cao X, Cheng K, Xiao W (2019) STEP-compliant CAD/CNC system for feature-oriented machining. Comput Aided Des Appl 16(2):358–368. http://cad-journal.net/files/vol_16/CAD_16(2)_2019_358-368.pdf
34. Yusof Y, Latif K (2016) New technique for the interpretation of ISO 14649 and 6983 based on open CNC technology. Int J Comput Integr Manuf 29(2):136–148. https://doi.org/10.1080/0951192X.2015.1030698
35. Noor H, Yusof Y, Kadir AZA, Latif K, Mohammed MA. A novel integrating between tool path optimization using an ACO algorithm and interpreter for open architecture CNC system. Expert Syst Appl 178, 114988. https://doi.org/10.1016/j.eswa.2021.114988

Authors and Affiliations

Kamran Latif[1] · **Efrain Rodriguez**[2] · **Renan Bonnard**[3] · **Yusri Yusof**[4] · **Aini Zuhra Abdul Kadir**[5]

Efrain Rodriguez
efrainrg2009@gmail.com

Renan Bonnard
renan.bonnard@gmail.com

Yusri Yusof
yusri@uthm.edu.my

Aini Zuhra Abdul Kadir
aini@mail.fkm.utm.my

[1] Department of Engineering Technology, Fakulti Teknikal Dan Vokasional, Universiti Pendidikan Sultan Idris (UPSI), Tanjong Malim, Perak, Malaysia

[2] Department of Mechanical Engineering, University of Brasilia, Brasília, Brazil

[3] Senai Innovation Institute For Embedded Systems, Florianópolis, Brazil

[4] Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Parit Raja, Malaysia

[5] School of Mechanical Engineering, Faculty of Engineering, Universiti Tecknologi Malaysia (UTM), Skudai, Malaysia