A Comprehensive Review on Part Programming Languages in CNC Machining

Sunil kumar, Manoj Kumar, Mohammed Suhaib, Nikhil Sharma, Trivendra Sharma, Sanjay Choudhary

Department of Mechanical Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur-302017 (INDIA)

Email: ***Sunil.kumar@skit.ac.in, Manoj.kumar@skit.ac.in, Suhaib.ansari@skit.ac.in, Nikhil.sharma@skit.ac.in, Trivendra.sharma@skit.ac.in, Sanjay.choudhary@skit.ac.in***

Received 06.01.2024 **received in revised form** 19.02.2024, **accepted** 09.04.2024

DOI: 10.47904/IJSKIT.14.1.2024.64-68

Abstract- This review paper comprehensively examines the landscape of part programming languages in the realm of Computer Numerical Control (CNC) machining. The paper delves into the historical progression of part programming languages, starting from the conventional G-code programming to high-level languages and contemporary approaches such as conversational programming and CAM systems. Additionally, the integration of CAD and CAM systems is explored, emphasizing the seamless translation of design specifications into CNC programs.

Keywords–. CNC, CAD, CAM, G-code programming

1. INTRODUCTION

Computer Numerical Control (CNC) machining stands as a cornerstone technology, enabling the precise and automated fabrication of diverse components. This introduction provides a contextual overview of the evolution and significance of part programming languages in CNC, setting the stage for an in-depth exploration of their historical foundations, contemporary applications, and future trajectories.

Historical Foundations: The roots of CNC machining trace back to the mid-20th century, coinciding with the advent of numerical control systems. Early on, the programming of machine tool movements was accomplished through manually generated codes, paving the way for the emergence of the foundational G-code language. As technology progressed, so did the complexity of machining tasks, giving rise to a spectrum of programming languages designed to address diverse manufacturing requirements.

Significance of Part Programming: Effective part programming is the linchpin of CNC machining, dictating the precise movements and operations executed by the machine tool. The choice of programming language profoundly influences the efficiency, accuracy, and adaptability of the manufacturing process. From the simplicity of G-code to the sophistication of high-level languages and Computer-Aided Manufacturing (CAM) systems, each language brings its unique attributes to the CNC programming landscape.

Objective of the Review: This review aims to provide a comprehensive understanding of part programming languages in CNC machining. It navigates through the historical journey of CNC programming, sheds light on the fundamental principles of conventional languages, and explores innovative approaches that redefine how manufacturers interact with CNC machines. By synthesizing insights from historical perspectives, current practices, and future trends, this review serves as a valuable resource for researchers, practitioners, and educators seeking to navigate the intricacies of CNC part programming.

Structure of the Review: The subsequent sections will delve into the intricacies of G-code and M-code programming, investigate the role of high-level languages in addressing complex machining tasks, explore user-friendly interfaces through conversational programming, and assess the integration of CAD and CAM systems. Industry standards, recent developments, challenges, and potential future trends will be critically examined, providing a holistic view of the dynamic landscape of part programming languages in CNC machining. Through this exploration, the review aims to contribute to a nuanced understanding of the pivotal role played by programming languages in shaping the future of CNC manufacturing.

2. LITERATURE REVIEW

Computer Numerical Control (CNC) machining has revolutionized modern manufacturing by enabling precise and automated fabrication of diverse components. At the heart of CNC machining lies part programming languages, which dictate the intricate movements and operations executed by machine tools.

Pandey et al. [8] conducted research elucidating the pivotal role of CAD/CAM Design, Development, and Manufacturing in modern manufacturing technologies. With processes growing increasingly complex, staying at the forefront requires top-tier

solutions capable of adapting to advancements while integrating intelligence across all discussed parameters.

Suk Hwan et al. [12] have effectively defined the user-friendly CAM software and presented the functionality of automatically generated three-dimensional tool path and imparting the CAD drawing.

Q.W.Wang et al. [2] provided a contextual overview of the evolution and significance of part programming languages in CNC, setting the stage for an in-depth exploration of their historical foundations, contemporary applications, and future trajectories.

HAN Y.L. [10] explained about macro programming that refers to the capability of creating custom commands or routines within part programs to automate repetitive tasks or to add specific functionalities.

Chen-Han Lee et al. [14] focused on CAD/CAM integration enhances collaboration, accelerates the product development cycle, minimizes production costs, and ensures the accuracy and quality of manufactured parts.

3. PART PROGRAMMING LANGUAGES

3.1 G-code programming

G-code commands are composed of alphanumeric codes, each serving a specific function in the machining process. Commands are organized into lines, where each line represents a distinct instruction. [5] The syntax typically includes a letter (indicating the type of command) followed by numerical values.

N: Line number for program control and reference.
G: Preparatory commands indicating the type of motion or operation.
X, Y, Z, A, B, C: Coordinates specifying tool or workpiece positions.
F: Feed rate, indicating the speed of tool movement.
S: Spindle speed, specifying the rotational speed of the cutting tool.
T: Tool selection for tool-changing operations.
M: Miscellaneous functions, such as turning the coolant on or off.
Some **G-Codes (Preparatory Commands):**
G0: Rapid positioning (non-cutting) movement.
G1: Linear interpolation for cutting movement.
G2, G3: Circular interpolation for clockwise and counterclockwise arcs, respectively.
G90: Absolute programming mode.
G91: Incremental programming mode.

3.2 M-code Programming in CNC Machining

While G-code governs the primary tool movements, M-code programming takes center stage in managing auxiliary functions crucial to the CNC machining process. This section delves into the intricacies of M-code, its syntax, and its pivotal role in orchestrating functions such as tool changes, spindle control, and coolant activation

M-codes are miscellaneous functions that extend beyond tool and workpiece motion. They control auxiliary actions such as spindle activation, tool changes, and coolant flow, contributing to the comprehensive orchestration of CNC machining processes. M-code commands follow a similar syntax structure to G-code. [3]

Some **M-Codes commands are:**
M0 and M1 halt program execution, providing a pause for operator intervention.
M2 and M30 signify the end of the program, triggering a program reset.
M3 activates the spindle in the clockwise direction, initiating cutting tool rotation.
M4 activates the spindle counterclockwise, allowing for reverse tool rotation.
M5 halts the spindle, ceasing tool rotation.
M6 is crucial for tool changes in CNC machining.
M8 activates the coolant system, initiating the flow of coolant during machining processes.
M9 deactivates the coolant, ceasing the flow.

3.3 High-Level Programming Languages in CNC Machining

In the dynamic landscape of CNC machining, high-level programming languages emerge as powerful tools for streamlining complexity and enhancing the efficiency of part programming. This section explores the characteristics, applications, and advantages of high-level programming languages, shedding light on their role in simplifying CNC programming.

High-level programming languages in CNC machining have evolved to address the increasing complexity of modern manufacturing tasks.

APT (Automatically Programmed Tool)

APT is one of the pioneering high-level programming languages designed for CNC machining.
It employs a symbolic language, allowing programmers to describe toolpaths and operations in a more natural and human-readable manner.

APT is capable of handling complex machining tasks, including contouring, drilling, and milling operations. It allows for the creation of toolpaths based on geometric features defined in the CAD model.[15]

APT introduced the concept of modularity, where subroutines could be defined and reused for similar machining operations. The language abstracts machine-specific details, making CNC programming

more accessible and less dependent on the intricacies of a particular CNC machine. APT is often integrated with Computer-Aided Design (CAD) systems, enabling a seamless transition from design to machining.CAD models can be directly translated into APT programs, reducing the need for manual intervention.

APT uses four statements viz. geometry, motion, post processor and auxiliary statements. Some APT commands are:

The GO/ command is used to initialize a sequence of contouring motions and may take alternative forms such as GO/ON, GO/TO, or GO/PAST.

C1 = CIRCLE/CENTER, P1, RADIUS, xx

CLPRNT is for cutter location print
UNITS/MM is used to specift units in mm or inches
PARTNO is used to identify the program. [5]

CLDATA (Cutter Location Data)

CLDATA, or Cutter Location Data, is another high-level programming language developed for CNC machining. It focuses on describing tool movements and cutter locations during machining operations. One of the key features of CLDATA is the separation of toolpath information from machine-specific details. This abstraction enhances portability, allowing the same CLDATA program to be executed on different CNC machines with minimal modification.

It is particularly well-suited for applications where the same part needs to be machined on different machines or with different tooling. Similar to APT, CLDATA provides a human-readable representation of toolpaths and machining operations. [7]

3.4 Computer-Aided Manufacturing (CAM) Systems: Transforming Designs into Reality

CAM (Computer-Aided Manufacturing) systems play a pivotal role in the modern manufacturing landscape, serving as the bridge between design concepts and the physical realization of components. CAM systems are software applications designed to facilitate the automated generation of CNC (Computer Numerical Control) programs from Computer-Aided Design (CAD) models. [6]
CAM software often includes simulation capabilities that allow users to visualize the material removal process. This helps in identifying potential issues, such as collisions or inefficient toolpaths, before the actual machining begins.

Advanced CAM systems support multi-axis machining, enabling the efficient programming of CNC machines with multiple axes of motion. This is particularly important for complex geometries.

CAM systems generate post-processed CNC code that is specific to the requirements of the target CNC machine. This code includes G-code and M-code instructions necessary for machining.

The integration of artificial intelligence (AI) and machine learning in CAM systems is an emerging trend, aiming to enhance toolpath optimization and decision-making processes. Cloud-based CAM solutions are gaining popularity, offering collaborative and accessible platforms for design and manufacturing teams. [14]

3.5 Conversational Programming in CNC Machining: A User-Friendly Paradigm

Conversational programming represents a paradigm shift in CNC (Computer Numerical Control) machining, offering a more intuitive and user-friendly alternative to traditional G-code programming. This approach simplifies the programming process by providing a conversational interface that enables users to interact with the CNC machine in a manner akin to a dialogue.

Conversational programming is an interface for generating CNC programs that involves interactive, step-by-step communication between the programmer and the CNC machine. [17] Conversational programming replaces the need for manually inputting G-code commands with straightforward prompts and questions. Users respond to prompts by entering values or selecting options, making the programming process more intuitive.

Conversational programming is well-suited for rapid prototyping and quick job setups. Operators can swiftly define toolpaths and machining parameters without extensive training in G-code programming.

Conversational programming may have limitations in handling highly complex geometries and intricate toolpaths. [13] It is most effective for straightforward and common machining tasks.

Different CNC machines may have variations in their conversational programming interfaces.

Users need to be familiar with the specific features and options available on their machines.

3.6 Macro Programming in CNC Machining: Enhancing Efficiency with Reusable Code Segments

Macro programming in CNC machining is a powerful technique that enables the creation of reusable and parameterized code segments. [8] This approach streamlines the programming process, enhances code modularity, and facilitates the development of efficient and standardized CNC programs.

Macro programming involves the creation of custom, user-defined programs or subprograms that can be invoked within the main CNC program.

Macros act as building blocks, encapsulating sequences of G-code commands to perform specific operations.

Macro programming has the ability to reuse code segments across multiple CNC programs. Macros support parameterization, allowing users to input variables that can be adjusted based on specific machining requirements. [9] This parameterization enhances the adaptability and versatility of the macro. Macro programming simplifies the main CNC program by replacing complex or repetitive sections with concise macro calls.

A macro for tool changes can be created, accepting parameters for tool number, tool length offset, and coolant status. This macro can then be called whenever a tool change is required.

3.7 Integration of CAD and CAM Systems: Streamlining Part Programming

The integration of CAD (Computer-Aided Design) and CAM (Computer-Aided Manufacturing) systems is a fundamental strategy in modern manufacturing, streamlining the part programming process. This seamless integration connects the design and manufacturing phases, facilitating a more efficient and error-resistant workflow. [12] This section explores the key aspects of CAD and CAM integration, highlighting its benefits and the collaborative synergy it brings to the production cycle.

The purpose is to bridge the gap between the virtual design of a component and its physical production by automating the generation of CNC (Computer Numerical Control) programs for machining.

A collaborative environment allows designers and machinists to work concurrently on the same project.

Real-time collaboration enhances communication, reduces lead times, and promotes a more iterative and agile design-to-manufacturing process.

Common file formats, such as STEP (Standard for the Exchange of Product Model Data) and IGES (Initial Graphics Exchange Specification), facilitate interoperability. [11]

CAD-to-CAM Workflow may be processed in following manner:

a. CAD Model Creation
b. Export to CAM
c. Toolpath Generation
d. Post-Processing
e. Machining Parameters and Instructions
f. Simulation and Verification
g. Post-Processing

4. CONCLUSION

G-M-code programming acts as the conductor for auxiliary functions in CNC machining, enhancing precision and operational flexibility. APT and CLDATA introduced abstraction and modularity, significantly impacting machining efficiency. Conversational programming streamlines CNC programming with user-friendly features, while macro programming facilitates systematic code reuse and parameterization. The integration of CAD and CAM systems revolutionizes part programming by enhancing collaboration and accelerating product development.

5. CHALLENGES AND TRENDS

There are some challenges and trends that were prevalent.

Cybersecurity Threats: The increasing connectivity of systems and the rise of IoT expose industries to cybersecurity threats. Protecting critical infrastructure from cyber-attacks remains a significant challenge. [17]

Supply Chain Disruptions: Global events, such as the COVID-19 pandemic, highlighted vulnerabilities in supply chains. Industries face challenges in creating resilient and flexible supply chain strategies.

Climate Change and Sustainability: Industries are under pressure to address environmental concerns. Meeting sustainability goals and reducing carbon footprints present challenges in areas like manufacturing, energy, and transportation.

Digital Transformation Complexity: While digital transformation brings efficiency gains, navigating the complexities of integrating new technologies, managing data, and upskilling the workforce poses challenges for many industries.

Regulatory Compliance: Industries must navigate a complex landscape of regulations. Staying compliant with evolving regulatory requirements, particularly in areas like data privacy and environmental standards, is challenging.

Workforce Adaptation: Rapid technological advancements necessitate a workforce with new skills. Bridging the skills gap and ensuring the workforce is adaptable to emerging technologies are ongoing challenges.

6. FUTURE SCOPE

Besides these challenges, there are some future trends reflect the ongoing evolution of industries as they navigate technological, societal, and environmental changes.

The landscape of technological innovation is poised for significant transformation across various

industries. Advancements in AI and automation, coupled with the proliferation of digital twin technology and edge computing, are set to revolutionize operational efficiency and decision-making processes. Additionally, breakthroughs in fields like quantum computing and biotechnology promise to tackle complex challenges, while renewable energy expansion and circular economy practices underscore a growing emphasis on sustainability. The convergence of human augmentation, 5G expansion, and decentralized finance further highlights the multifaceted evolution shaping the future of technology and industry.

REFERENCES

[1] Juan Zhang, Research on CNC Lathe Programming and Improving Machining Accuracy, IOP Conf. Ser.: Mater. Sci. Eng. 452 042050,2018

[2] Q.W. Wang, X.W. Xu: Discussion on the Simple Skills of CNC Lathe Programming [J], (2011) No.8, p: 178. (In Chinese)

[3] Z. H. Yu: Thoughts on the Programming and Operation of CNC Lathe [J], (2015) No.11, p: 299. (In Chinese).

[4] Sutarman, Haryono Edi Hermawan, Sarmidi, Computer Numerical Control (CNC) Milling and Turning for Machining Process in Xintai Indonesia, Journal of Research in Mechanical Engineering Volume 3 ~ Issue 5 (2017) pp: 01-07

[5] Mikell P. Groover, Automation, Production Systems, and Computer-Integrated Manufacturing, Fourth Edition, Pearson publication

[6] Zoya Rizvi, Adeel Khan, Parshva khatadia, Devang Jadhav, Daksh Bafna, A study of Computer Numerical Control (CNC) toolpath and profile milling on CAM software, IJCRT | Volume 9, Issue 2 February 2021

[7] Anuradha, N et al (2017) Design and Manufacturing of Flange Coupling Using CNC Technology. International Journal for Research in Applied Science and Engineering Technology. ISBN: 2321-9653. Vol 5. Issue 8

[8] Rohit Pandey, Arvind Singh Tomar & Nishant Sharma (2016): "A Recent Role of CAD/CAM in Designing, Developing and Manufacturing in Modern Manufacturing Technologies" Imperial Journal of Interdisciplinary Research (IJIR), 2(3),399.

[9] WANG F. B, SUN S.B. (2011) Application of Macro Program Aiming at FANUC System to Parabolic.Coal Mine Machinery, vol. 32 No.10:148–150.

[10] HAN Y.L. (2008) Application of the Cam Machining on the Macro Program.Coal Mine Machinery, Modern Manufacturing Engineering, No.5:32–33.

[11] Z. Gal-Tzur, M. Shpitalni and S. Malkin, "Design and Manufacturing Analyses for Integrated CAD/CAM of Cams". J.Eng. Ind 111(4), 307-314, 1989, 8 pages (Revised 2008).

[12] Suk-Hwan Suh, Sung-Kee Noh, Yong-Jong Choi, (1995),"A PC-based retrofitting toward CAD/CAM/CNC integration", Computers & Industrial Engineering, 28 (1), 133-146.

[13] Bohlen, I. S., Fieret, J., Holmes, A. S., Lee K. W., 2003, CAD/CAM Software for an Industrial Laser Manufacturing Tool, Photon Processing in Microelectronics and Photonics Ii, 4977 198-206.

[14] Chen-Han Lee, Lingyun Lu, Jon Dym and Guangyan Yin (2004): "Shape Preserving Global Parameterization for CAD/CAM/CAE", International Mechanical Engineering Congress and Exposition Manufacturing Engineering and Materials Handling Engineering Anaheim, California, 237-249.

[15] Rishi Kumar Shukla, Dinesh, B. Deshmukh (2015) "A Review on Role of CAD/CAM in Designing for Skill Development", ITMVU, at Vadodara, with 1,827 Reads

[16] Horath, Larry, "Computer Numerical Control Programming of Machines", Macmillan, NY, 1993

[17] M.N. Latif, R.D. Boyd, R.G. Hannam: "Integrating CAD and manufacturing intelligence through features and objects", I. J. Computer Integrated Manufacturing, Vol. 6, No. 1 and 2, pp 87-93, 1993.