



An Object Oriented Approach to the Online Simulation of CNC Turning and Milling Machining Operations through Computational Geometry

Vikas K. Malviya and Prashant K. Jain

PDPM Indian Institute of Information Technology, Design and Manufacturing Jabalpur
ervikasjec@gmail.com, pkjain@iiitdmj.ac.in

ABSTRACT

This paper presents an approach to the simulation of the CNC turning and milling operations. An object oriented approach is selected as it is perfectly suitable for the development of simulation system. In that approach the problem can be decomposed into objects that are real time entities. Object oriented analysis and design (OOAD) is used for the analysis of problem and the design of the solution in terms of objects. OOAD uses object-oriented perspective to model a system as group of interacting objects that work together to perform required functionality. The analysis has been performed and the modeling has been done with Unified Modeling Language (UML). For the verification of the analysis and design, online simulation software for the CNC turning and milling operations has been developed. The simulation facility is provided not only for simpler codes but also for the codes with canned cycles. For the simulation of various canned cycles present in part program, the software first simplifies them into simpler codes without canned cycles and then simulate it. The simplified program obtained from the software can be saved by the user and can be used on other machines that don't have the facility to run canned cycles. For the development of the software Java platform has been selected. Key technologies used for the development of software are JSP (Java server pages), Java applet and HTML (Hyper text markup language). Simulation software is available online at the URL "<http://virtualcnc.iiitdmj.ac.in>".

Keywords: object oriented approach, Java, canned cycles.

DOI: 10.3722/cadaps.2013.839-849

1 INTRODUCTION

Advanced manufacturing technology is the modern approach for producing automated, customized computerized designs. The primary aim of advanced manufacturing technology is to produce high quality products with low cost in the shortest time. Numerical control technology is an important part of advanced manufacturing technology. Numerical Control (NC) is the control of a system or device by direct input of data in the form of numbers, letters, symbols, words, or a combination of these forms. Numerical control enables the user to operate machine tools with the use of numbers and symbols. Computer Numerical Control (CNC) brought tremendous changes to the manufacturing industry as complex structures became relatively easy to produce, and the number of machining steps that

required human action has been dramatically reduced. CNC machines reduced the frequency of errors and provided CNC operators with time to perform additional tasks. Everything that an operator would be required to do with conventional machine tools is programmable with CNC machines. The ever-increasing use of CNC in industry has created a need for personnel who are knowledgeable about and capable of preparing the programs which guide the machine tools to produce parts to the required shape and accuracy. Training of persons for that required level is so costly and has problems of risk of equipment. In this scenario online simulation system for CNC turning and milling operation can be a great tool for training.

In a previous attempt Hanwu and Yueming (2009) proposed a web based virtual operating system for operation training for manufacturing facility and manufacturing process simulation. They have used Virtual Reality Modeling Language (VRML) for simulation. VRML is a file format for describing interactive 3D graphics. VRML lacks the capability of reading and writing of files. To overcome that problem author has used the combination of VRML, Java Applet and Java Script. VRML connects with Java Script that in turn connects with Java Applet to perform simulation [6]. Masood et al. (2002) has also discussed techniques used in the simulation and control of PROTEC computer numerically controlled (CNC) machine software. This CNC control system developed in Visual Basic works on Windows 95 or later. Simulation of the machining operation is mainly based upon interpolation of tool path. They have used Digital Differential Analyzer (DDA) method for performing linear interpolation and circular interpolation [12]. Shuang et. al. (2008) presented virtual NC simulation as virtual reality technology as an approach to overcome the shortcomings of traditional training like high training cost and high risk of equipment. Open Inventor and Visual C++ 6.0 were taken as the development platform of virtual machining system. Open Inventor is an object-oriented 3D toolkit developed by TGS Corporation. It simplifies the procedure for graphic design. The simulation system consists of five parts that are, NC compiler, virtual machining environment, conduct visual simulation, human-computer interaction and authentication [14]. Arthaya et. al. (2010) designed and developed a kind of program checker to see whether the program for a CNC turning machine is correct and ready to be fed to the machine. This program was also capable of displaying cutting simulation of how the product to be cut in the machine. This program was specifically developed for a CNC Okuma Howa ACT 3 turning machine with Fanuc 6T controller and it can be extended to other machine types. Software was developed with C++ programming language [1].

In the present work an object oriented approach for the simulation of CNC turning and milling machining operations is presented. The approach is modeled through the use of various UML diagrams. For the practical implementation of the approach, an online system is developed. It not only simulates the code but also simplifies the complex canned cycles by converting them into simpler codes. That simplified code can then be stored by the user in form of a text files.

2 SYSTEM ANALYSIS DESIGN AND MODELLING

Object oriented modeling and design is a new way of thinking about problems using models organized around real-world concepts. The fundamental construct is the object, which combines both data structure and behavior in a single entity. Object-oriented models are useful for understanding problems, communicating with application experts, modeling enterprises, preparing documentations, and designing programs and databases.

In this process, first an analysis model is built to abstract essential aspects of the application domain. This model contains objects found in the application domain, including a description of the properties of the objects and their behavior. Then design decisions are made and details are added to the model to describe and optimize the implementation. The application-domain objects form the framework of the design model, but they are implemented in terms of computer-domain objects. Finally the design model can be implemented in a programming language, database, or hardware.

Superficially the term "object-oriented" means that we organize software as a collection of discrete objects that incorporate both data structures and behavior. This is in contrast to conventional programming in which data structures and behavior are only loosely connected. There is some dispute about exactly what characteristics are required by an object-oriented approach, but they generally include four aspects: identity, classification, polymorphism, and inheritance.

Software design is the most important step in the process of software development. Without good design a software development project may be likely to struggle, cause frustration or can fail. This is true for team development projects as well as for smaller one man software projects. The software development starts with the requirement analysis and design these are two most important phases of the software development process. In the present work the analysis of the machines were carried out and based on that analysis the modeling of the system is performed. The modeling of the system is done with the help of unified modeling process. UML is the language that visually specifies the artifacts of the system. It is language independent and used just for modeling the system. UML provide different diagrams to help software engineer to analyze and design object-oriented software. UML diagrams represent structure, behavior, and interaction of and between classes, objects, and components [9].

Fig. 1 describes the use case for the developed system. It clarifies that the user can create new file, open an existing file and edit it and then save it on the system. The execution use case depends on other use cases like debug use case and the execution use case. The execution use case consists of two other use cases that are simulation and analysis. Analysis of part program consists of the calculation of machining time and length traveled by the tool.

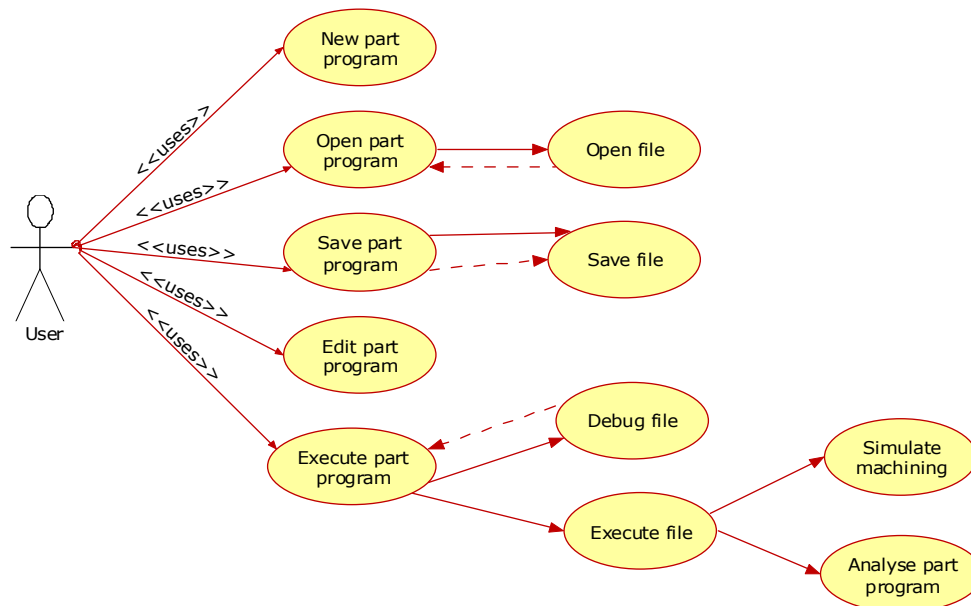


Fig. 1: Use case diagram for the system.

Use cases are used only for the specification of the system. It does not provide information about the order in which the operations are taking place. For showing the sequence in which the user is interacting with the system is given through the sequence diagram. UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system.

The sequence diagram for the developed system is shown in

Fig. 2. The system is broken down in four parts and the sequence diagram shows the interaction between these parts. The main window of the system starts the interaction with the user and it has options to create new file, open existing file, edit file and finally execute or run the part program to simulate the machining process.. The other parts of the system are NC code handler, file handler and simulator.

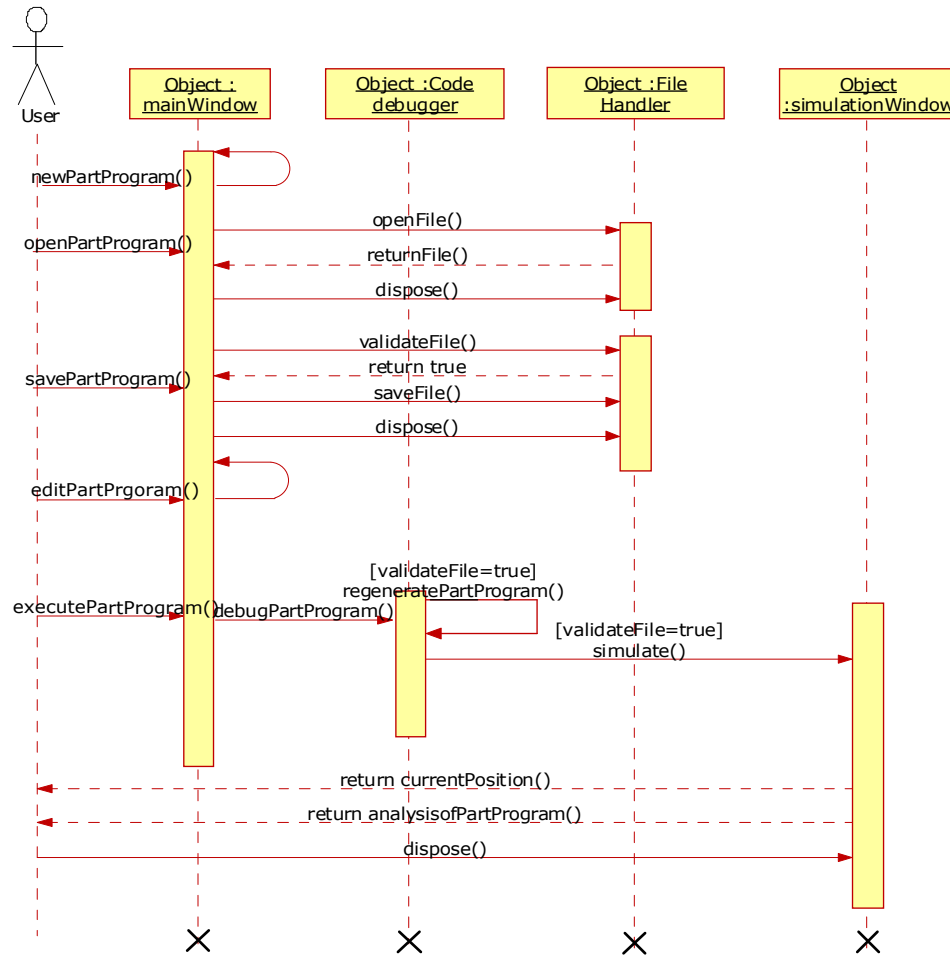


Fig. 2: Sequence Diagram for the system.

CNC machines work through the use of a computer that reads a particular code called a G-M Code (Part Program) to perform certain operations. The numbers, letters, and symbols forms the coded instructions that refer to specific distances, positions, functions, or motions, that the machine tool can perform as it machines the work piece. Besides these simple G and M codes controllers of modern machines also contain canned cycles. Canned cycles are specialized and predefined machine instructions permanently stored in the controller of the machine. These are used to perform machining operations that are of repetitive nature. Canned cycles are initiated by a single G-code. Whenever the code for these canned cycles encountered, the controller of the machine calls these instructions and executes. The use of canned cycles reduces length of code, efforts and time spent in programming and simplifies the programming process [15], [16] as they can be accessed with a small block of part program rather than with many individual tool movement instructions [13]. The ability to alter program is one of the greatest assets of using canned cycles. For utilizing canned cycles one has to provide the profile points only. The calculation of intermediate points is performed by the machine itself. Canned cycles control operations like drilling, boring, spot facing and so on with milling control.

The NC code handler in the developed system is consists of two parts that are code debugger and code regenerator. The working of code debugger and code regenerator is described with the use of activity diagrams. Activity diagram is basically a flow chart to represent the flow form one activity to

another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.

Fig. 3 describes the activities performed in the process of debugging the part program. Debugger reads part program line by line. Whenever the specific keyword comes the control jumps to checking algorithm written for that word. If the check fails for the valid code then the error value is returned for that line.

G-codes are divided in three types for the debugger routine. These are

- **Functional G-codes:** These codes include the g-codes that are having some parameters or that perform the movements. Interpolation and other codes like dwell are classified into this category. These type of g-codes are handled by separate functions written for them as they have many parameters with them so they are checked by a different functions. Whenever the codes encounters the control goes to the particular function. If error occurs then the object of the error class with the specific error message and the line in which error is returned, and displayed to the user.
- **Non-functional G-codes:** These codes include the g-codes that set some conditions for the part program. Codes used for setting of incremental or absolute mode, setting of mm or inch for measurements are classified in this category. For debugging this type of codes there is no need of a separate function as they set only some conditions. Debugging routine checks these functions as they come and sets the conditions according to them.
- **Canned cycle G-codes:** These codes include the g-codes that represent a canned cycle. These are handled by their separate debugging functions. These codes have a large number of parameters so the debugging routine for these codes are separate from other functions. The debugging function checks for the parameters of the cycles and their appropriate values. If the error occurs then the object of error class is created and returned to the main control.

Separate functions have been written for the validation and processing of different g-codes. Whenever a particular g-code occurs in the program, control automatically jumps to function that validate that g-code and checks all the parameters like dimensional words (e.g. X(U), Y(W), Z etc.) necessary to validate that g-code. Also specific words like I, J, K, etc. are also a part of specific g-code functions so they are also checked by the g-code validating functions. The part program given by user contains not only simple but also canned cycle codes. The cycle code is simplified and converted into simpler codes. After simplification the part program is regenerated and N codes associated with each line of part programs entered by user get changed. So the routine inserts the N-code for the blocks and give the program to user.

An activity diagram for code regeneration is shown in

Fig. 4. The routine checks each line of part program. If canned cycle occurs the control goes to the specific class of canned cycle. In the work the canned cycle is represented with a separate class that includes the cycle parameters as the member variables of the class and the functions to simplify cycle as the member functions of the class. When the calculations are performed the control again goes to the main program. The simplification procedure includes the calculation of intersection and intermediate points between the profile points as described in the next section.

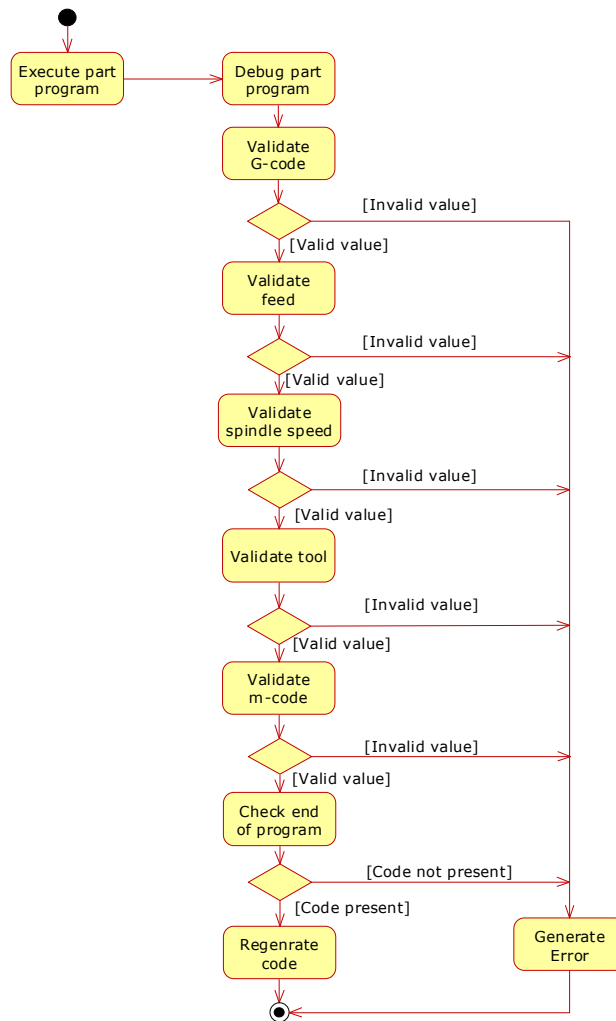


Fig. 3: Activity diagram for debugger routine.

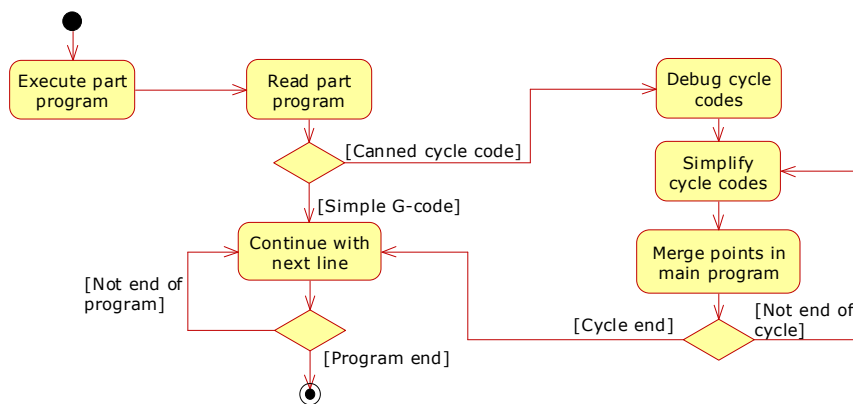


Fig. 4: Activity diagram for code generation.

3 IMPLEMENTATION OF THE SYSTEM

Implementation of the system includes two steps. First is the simplification of canned cycles and the other is the simulation of the machining process performed by the simplified part program. It is worthwhile to discuss working of a canned cycle to understand the simplification process developed in the present work. Here multiple facing cycle is presented for demonstrating the working of canned cycle.

Multiple facing cycle removes the metal along the X-axis direction. For using this cycle the final geometry of the work piece is defined and given in the different blocks of the cycle

Fig. 5(c). Then the tool path is calculated automatically by machine controller as shown in

Fig. 5(a), (b), and material is removed in successive passes.

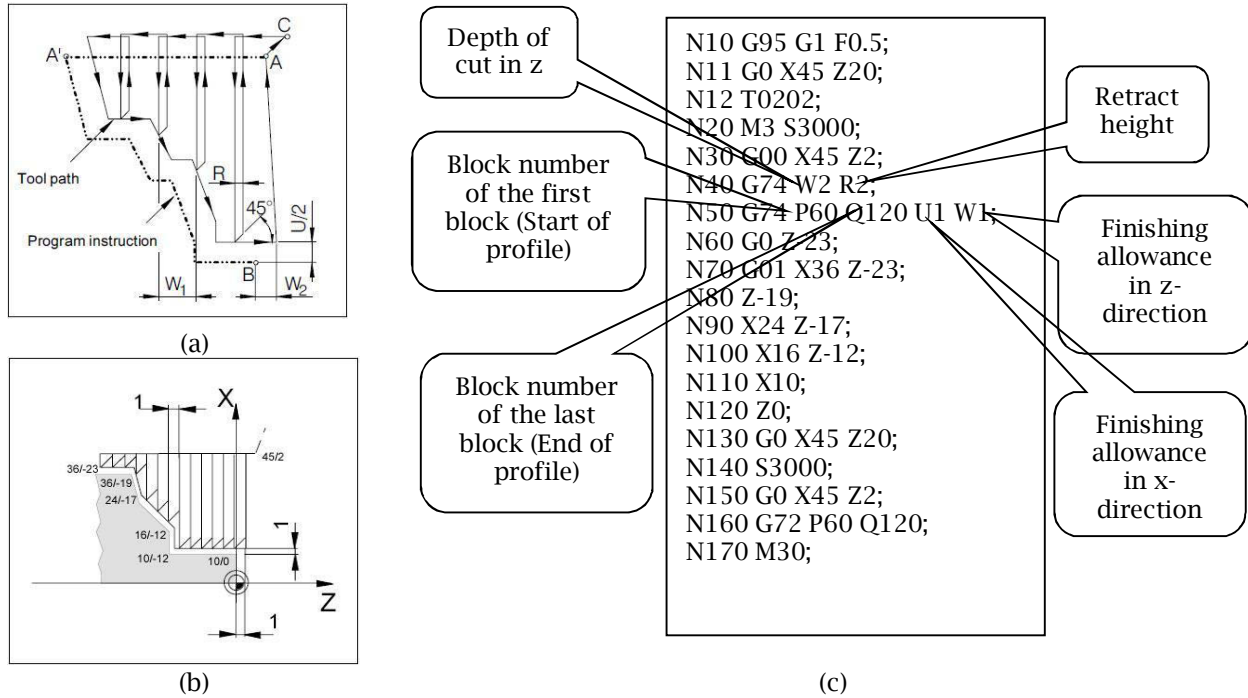


Fig. 5: Multiple facing cycle, (a) Nomenclature, (b) Tool path, (c) Sample part program and description of cycle parameters [2], [3].

The simplification procedure is based on the calculation of intersection points of tool path with the work piece profile/geometry. The intersection points (*) (as shown in Fig. 7(b)) are calculated on the basis of type of profile to be cut, i.e. linear or circular. Then intermediate points () (as shown in Fig. 7(b)) to complete one cut are found out and arranged in a logical sequence with the calculated intersection point. The simplification procedure is given in Fig. 6 in the form of a flowchart. Calculation of intermediate points depends on the canned cycle. In case of multiple facing cycle tool path is always parallel to the x-axis therefore a line parallel to x-axis is considered and offset every time towards the desired profile by the value of depth of cut (d) mentioned in the part program, to calculate intersection point with workpiece profile as shown in Fig. 7(a). As shown in the diagram, intersection point of the line representing tool path and the segment representing the profile to be cut, i.e., profile connecting points (x_1, z_1) and (x_2, z_2) is (x, z) . Calculation of intersection point is different for different geometry (line or curve) to be cut as shown in Fig. 7(b). Fig. 7(c) shows the sample code for calculation of intersection points successively.

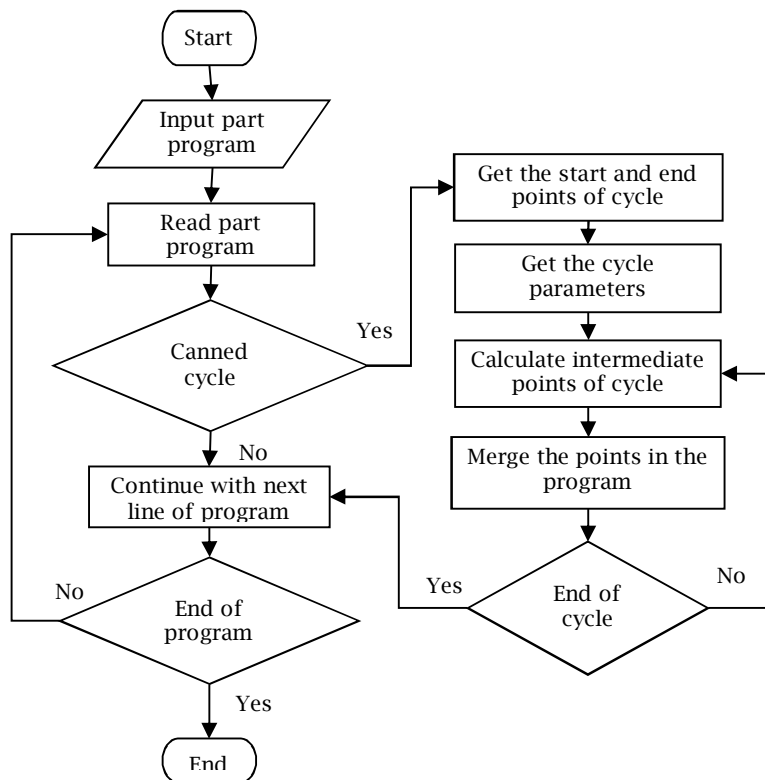


Fig. 6: Flow chart for simplification procedure.

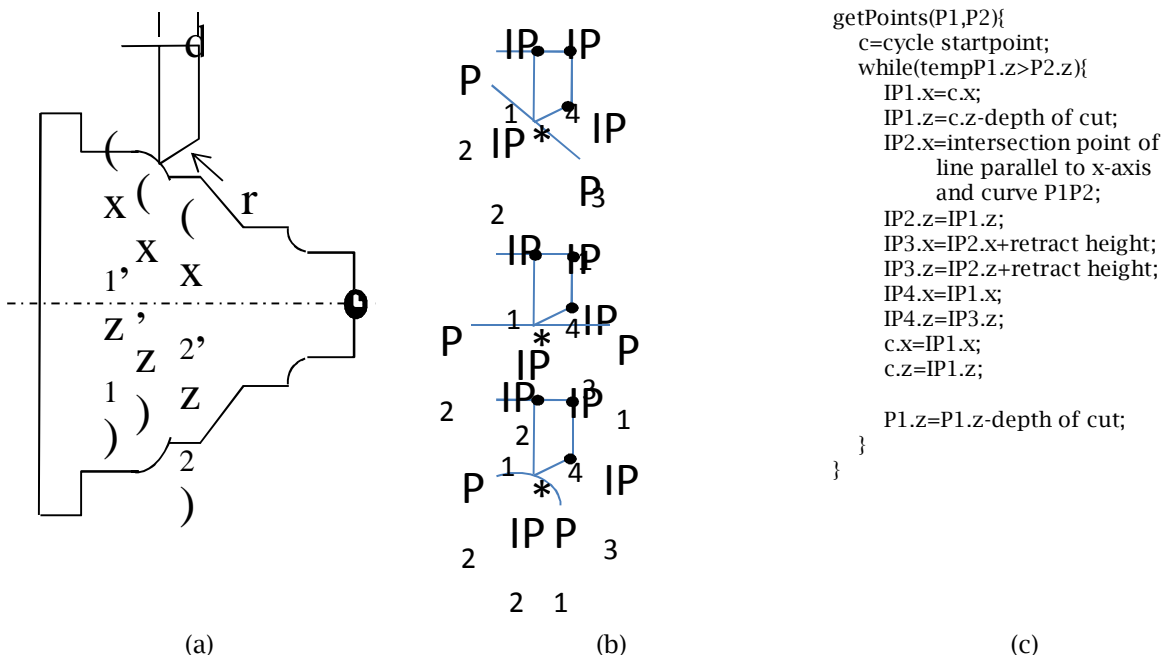


Fig. 7: Procedure for calculation of intermediate points in multiple facing cycle: (a) Work piece profile, (b) Types of profiles, (c) Sample code for getting intersection points.

Java platform has been selected as the development platform of the system. Java provides a beautifully designed set of classes for high quality graphics and imaging. Java is a very efficient language for applications in graphics and visualization with the benefit of portability over a wide range of platforms [4]. Further Java is also suitable for web based distributed environments with high performance using Just in Time (JIT) compiler [5]. One of the benefits of using the Java platform is the extensive support for networking, multithreading, multimedia, database access and web services.

The developed software performs debugging of the NC code, regeneration of the code, simulation of machining process and analysis of part program. The regeneration of code is performed by the simplification of complex canned cycles by converting them into simpler codes. This regenerated part program is returned to user in the form of a text file. User can now save that part program to run on other machines that don't have the facility to run canned cycles.

The developed software is available for users offline as well as online. For offline version user need to download the setup that is available on the web page. For using online system user has to go to web page of the software. The online system is developed with JSP, Java Applet and HTML (Hyper text markup language)

Implementation of the system is carried out keeping in view the three types of options to use the system. The three options by which the user can use the system are:

- The system is available as an executable setup that will install the whole system on the user's computer. The setup is available on the project web site. This scheme is simple and user is familiar with it. The drawback of this approach is that user has to check for updates manually.
- The system is available online. User can go to project web page where the online editor is available that can be used for the debugging, simplification of the part program and for simulation of the part program.
- In this approach user has no need to install the system on their computer. JWS is used for the implementation. In this approach there is no need to download and install JRE, no need to remain connected to internet for executing program. Updates are downloaded automatically whenever they are available. A limiting factor for this approach is the size of the file to be downloaded. With Pack200 a large jar file can be compressed to one-ninth of its original size. Pack200 is the compression format introduced to decrease the size of java files [7], [8].

Some screens of the developed user interfaces have been presented here to describe the functionality of the developed software. Fig. 8(a) shows the home page of the application which shows various links, welcome screen, the structure of the part program and important instructions to use the software. Fig. 8(b) shows the online editor web page which provides the facility to write, debug and edit the part program. Any error occurred during debugging of part program will be displayed on this page. Fig. 8(c) and Fig. 8(d) shows the simulation of milling and turning operations respectively in the developed software. Current position of the tool and assigned process parameters are also displayed during the simulation.

4 RESULTS AND DISCUSSION

The approach for the simulation of CNC turning and milling machining operations based on object oriented analysis and design is presented and successfully implemented. In the present work first the analysis of the machines is carried out and then the designing of the full software system is performed. The approach is presented here with the use of UML diagrams. The benefit of that approach is that object oriented languages are currently popular as they models the complex software systems in the form of objects and the interaction among them. Hence a real world can be easily modeled by these languages. The presented approach provides a framework that can be used by other languages and can be extended easily.

The developed system performs the calculation involved in canned cycles and simplifies the canned cycles into simpler programs and simplified program can be obtained using developed system as a text file. Developed system generates simplified part program in a text file and returns to the user. Now by using this part program a complex shape can be machined even on machines that don't have the facility to run the canned cycles. This will provide great convenience to the users of low cost machines with very less facilities as user can write part program with canned cycle despite the fact that

machine does not have facility to run canned cycles. This has been demonstrated using case studies. [10],[11]

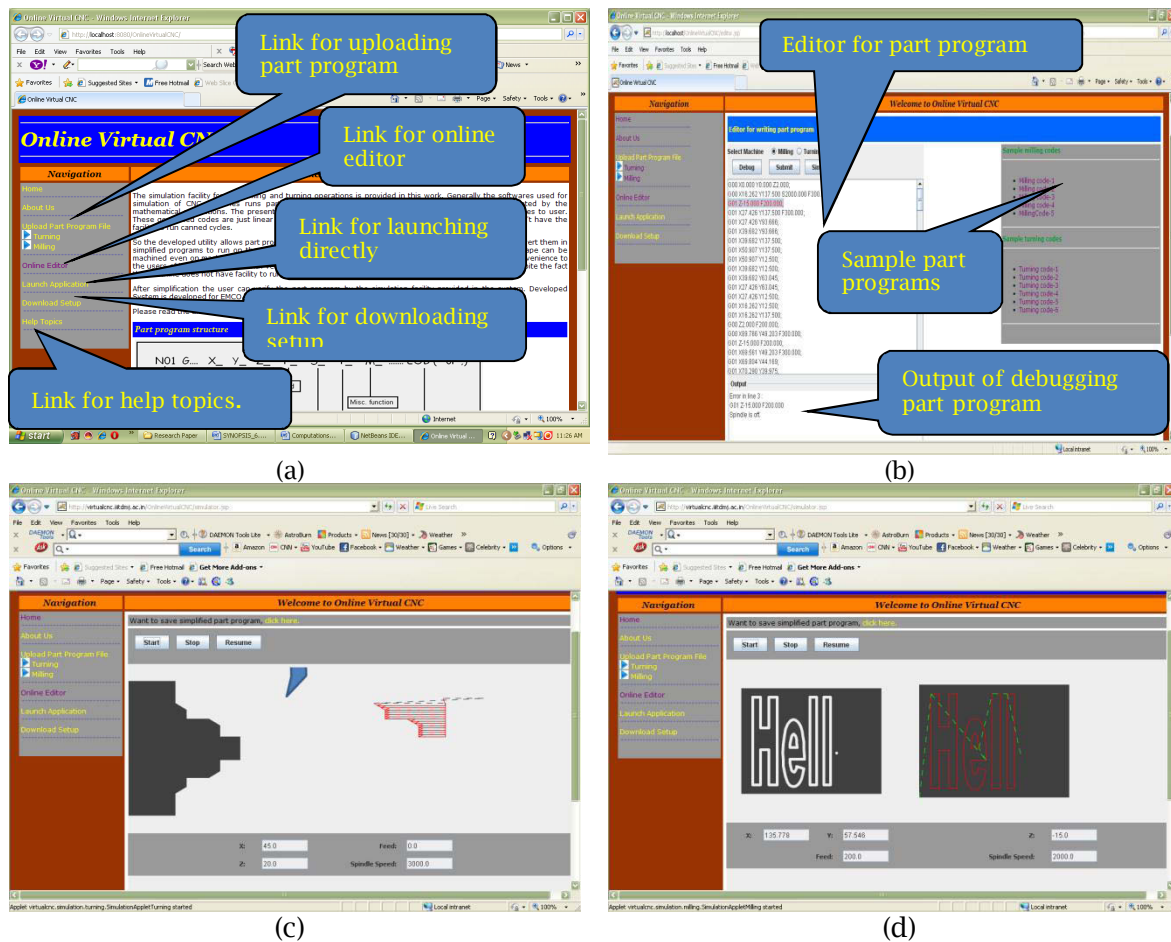


Fig. 8: Some screens of the developed system.

The developed simulator provides an environment for the training and learning of CNC machines. The cost constraint and the risk of equipment are the main restrictions for the training and learning the CNC machines. The proposed work addresses these problems and provides an easy to use environment. Developed software provides the facility to online write, debug, edit, simulate and analyze part program. It provides an easy to use interface even for a novice user to learn the complex working of CNC machines. The system can be used to train the people to work on CNC machines. The developed simulation software is also useful to teach students about the working of CNC machines and the complex canned cycles.

REFERENCES

- [1] Arthaya, B.; Setiawan, A.; Sunardi S: The Design and Development of G-Code Checker and Cutting Simulator for CNC Turning Operation, Journal Of Mechanical Engineering Research, 2(3), 2010, 58-70.
- [2] EMCO WinNC GE Series Fanuc 21 TB Manual.
- [3] EMCO WinNC GE Series Fanuc21 MB Manual.

- [4] Evai, F.-D.; Rahman, M.-M: The Suitability of the Java Language for Computer Graphics and Visualization, Proceedings of the Sixth International Conference on Information Visualization (IV'02).
- [5] Geoffrey, F.; Wojtec, F: Towards Web/Java-Based High Performance Distributed Computing-an Evolving Virtual Machine, IEEE Proceedings of NPDC-5' 96, 1996, 308-317.
- [6] Hanwu, H.; Yueming, W.: Web-Based Virtual Operating of CNC Milling Machine Tools, Computers in Industry, (60), 2009, 686-697.
- [7] Java Web Start Tutorial,<http://download.oracle.com/javase/tutorial/deployment/webstart/>.
- [8] Java Web Start, http://www.java.com/en/download/faq/java_webstart.xml
- [9] Khan, W.-A.;Raouf, A.;Cheng, K.: Virtual Manufacturing, Springer, 2011. <http://dx.doi.org/10.1007/978-0-85729-186-8>
- [10] Malviya, V.-K.; Jain, P.-K.: Computations for Simplification of Canned Cycles in CNC Programming for Milling Operations, Proceedings of International Conference on Computational Methods in Manufacturing (ICM2011), IIT Guwahati, India, 2011.
- [11] Malviya, V.-K.; Jain, P.-K: Computations for Simplification of Canned Cycles in CNC Programming for Turning Operations, Advanced Materials Research 383-390, 2012, 965-971. [doi:10.4028/www.scientific.net/AMR.383-390.965](http://dx.doi.org/10.4028/www.scientific.net/AMR.383-390.965).
- [12] Masood, T; Nlubashar, A.; Khan, J.-M.-A.: Techniques used in the Simulation and Control of Protec Computer Numerically Controlled [CNC] Machine Software, IEEE ICIT, Bangkok, Thailand, 2002, 504-509.
- [13] Mattson M.: CNC Programming Principles and Applications, Delmar Cengage Learning, 2010.
- [14] Shuang, S.; Lufeng, L.; Guoqin, L; Xiangjun, Z; Jingfeng, Y.: The Virtual Simulation System Of Numerical Control Machining, International Workshop on Modelling, Simulation and Optimization, 2008, 313-318.
- [15] Smid P.: CNC Programming Handbook, Industrial Press Inc., New York, 2003.
- [16] Stenerson, J; Curran K.: Computer Numerical Control-Operation and Programming, Prentice Hall, 2005.