International Journal of Computer Integrated Manufacturing, 2015
http://dx.doi.org/10.1080/0951192X.2015.1030698

Taylor & Francis
Taylor & Francis Group

New technique for the interpretation of ISO 14649 and 6983 based on open CNC technology

Yusri Yusof* and Kamran Latif

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

(Received 3 July 2014; accepted 18 January 2015)

Computer numerical control (CNC) controller is the termination organiser, which is located at the end of the manufacturing process chain. The CNC controller is composed of two modules: software and hardware. The software module of the controller is known as interpreter that translates International Standards Organization (ISO) data interface model instructions to the hardware parts of the controller in the required structure of the machine. The aim of next-generation CNC is to be portable, interoperable, flexible, open, intelligent and adoptable. In the progression towards the development of next-generation CNC systems, the problems of current ISO data interface model limitation and CNC machine vendor dependency were faced. In order to overcome these issues, the ISO standard (ISO 14649) and open architecture control (OAC) technology were introduced. In this paper, a new technique for the interpretation of both ISO data interface models has been presented. The developed module is specially designed for the OAC-based CNC systems that translate the ISO 14649-21 and ISO 6983 information to the CNC machine in required structure. The module also provides a shop floor editing facility and is also able to generate the physical file output in user-defined structure of .txt and .xml formats.

Keywords: CNC; STEP-NC; interpreter; CAD/CAM; open architecture control; LabVIEW

1. Introduction

In the 1950s, a revolutionary change occurred in the manufacturing world with the introduction of numerical control (NC) machine tools. NC is the term used to define the control of machine movements with other various functions that transfers the instructions in a series of numbers via an electronic control system (Liang, Hecker, and Landers 2004). The next significant development towards machine automation occurred in the 1960s with the introduction of direct NC. This system allows the programmers to send program files directly to the machine unit via central control system (Ramesh, Jyothirmai, and Lavanya 2013). Later, in the 1970s, a major revolutionary change occurred in machine tool automation with the development of the computer numerical control (CNC), where a computer replaced most of the electronic hardware and punch cards of the NC machine (Liang, Hecker, and Landers 2004). The CNC machine works on the technology that uses minicomputers to generate, parse and execute the sequential control. These CNC machines are composed of minicomputers and computer-aided design (CAD) drawing software to support the development of on-machine programs to enable machining of different parts (Talavage 1987). These CNC machines are classified into different categories like machining process, number of axes, spindle arrangement, number of spindles and kinematics configuration (Stenerson and Curran 2005; Vichare et al. 2009). The application of this technique is often used in turning, drilling, milling, electronic component insertion, tube welding and cutting robots (Groover 2007).

In the progression towards more modern systems, flexible manufacturing became dominant in the 1970s and the 1980s to enable low-batch production of a wide range of parts. In order to realise flexible manufacturing, CNC machines became a critical manufacturing resource due to their capability for being reprogrammed to produce different parts (Xu and Newman 2006). CNC technology has proved to be economical in mass, batch and many other single-item production cases. Some of the most important factors which contribute towards the economic feasibility of CNC technology are high productivity rates, uniformity of the product, reduced component rejection, reduced tooling costs, less operator involvement and easy machining of complex shapes (Yusof, Kassim, and Zamri Tan 2011).

Today in industries, many CNC machines are employed with different controllers and multiple abilities to fulfil customer demands. These CNC machines are composed of many parts; the controller is one of the parts that has two modules inside: hardware module and software module. The function of the software module is to translate the input International Standards Organization (ISO) data interface model code to the internal structure of the machine. CNC controller reads the ISO data interface model instructions, interprets them and performs numerically direct interpolation of the cutting tool in the workpiece (Ertell 1969). The machine controllers interpret the

---

*Corresponding author. Email: yusri@uthm.edu.my

part program to control the tool motion with multiple axes. The interpreter of the CNC system is the software module of the controller unit that translates the part program consisting of code commands and related addresses into internal commands for moving tools and executing auxiliary functions in a CNC system.

The CNC machines are programmed in ISO 6983, formally known as G&M codes language (ISO 6983-1 1982). The current CNC language, the so-called G&M codes, has been used since the 1950s. The programs for the CNC machines are generated by computer-aided manufacturing (CAM) systems that use CAD information. The G&M codes language is defined by just numerical codes such as G, T, M, F, S, etc., indicating the movement of a machine and an axis to the controller. Meanwhile, this language delivers only limited information to the CNC, which makes the CNC nothing but an executing mechanism which is completely unaware from the motions being executed. However, with the growth in the manufacturing world, there are a number of problems found in this data interface such as delivering limited information to CNC, transferring one-way information from CAD/CAM to CNC, unable to implement the seamless integration of the CAD–CAM–CNC, etc. (S. H. Suh and Cheon 2002). Also, the development of various CAD/computer-aided process planning (CAPP)/CAM/CNC systems in the CAx process chain created a requirement for data exchange standards to enable smooth data integration between machines and systems from different vendors. Behind these drawbacks in reality, G-code programs are still very valuable in commercial manufacturing systems because they incorporate both an implicit microprocess plan and many years of operator experience (Shin, Suh, and Stroud 2007). These codes are well accepted worldwide and are still in use; for this reason, a functionality of G-code interpretation has been included in this system.

In order to overcome interoperability issues, a Standard for the Exchange of Product Data (STEP) was introduced in the 1990s (ISO 1991; I. ISO 1994). The objective of STEP is to provide a means of describing product data throughout the life cycle that is independent from any particular computer system. The initial release of STEP was published in 1994 by ISO, in which parts 1, 11, 21, 31, 42, 43, 44, 46 and 101, and application protocols (AP) 201 and AP 203 were introduced (T. ISO 1994). Later in 2002, AP 202, AP 209, AP 214, etc., were published by ISO; these APs had enabled more skills in the STEP for different industries. With the development of various STEP parts and APs, some problems emerged with these large data structures; these problems were rectified with the introduction of the STEP modular architecture (Houshmand and Valilai 2012; Feeney 2002). Currently, AP 242 is being developed, which is based on geometric dimensions and tolerances in combination with STEP AP 203 and 214 (Safaieh, Nassehi, and Newman 2013). ISO 10303 significantly improved the interoperability between CAD systems. This standard has also highlighted the need for the development of a similar standard for exchange of information between CNC machines as well as CAM systems. Consequently, in 1999 an international project was started to specify a new standard entitled ISO 14649, formally known as STEP-NC, to bring the benefits of STEP to CAM and CNC (S.-H. Suh, Cho, and Hong 2002). The ISO 14649 standard is an extension of ISO 10303, which allows the connections between STEP-based CAx and CNC. This standard is still in the developing phase under two different subcommittees (SC) of Technical Committee (TC) 184 under ISO: SC1 and SC4. ISO 14649 is under SC 1 of TC 184 that focused on the machine control. This model is for CNC known as application reference model, whereas ISO 10303-238 is under SC 4 of TC 184 that focused on the industrial data. This model is for application interpreted model for CNCs (Feeney et al. 2003; Hardwick et al. 2013). Actually ISO 10303-238 is an upgradation of ISO 14649; the difference between them is highlighted in Hardwick et al. (2013) and Krzic, Stoic, and Kopac (2009).

The STEP-NC code contains high level of information. It has two sections: 'HEADER' and 'DATA'. The 'HEADER' section contains the information of file name, author, date, etc., whereas the 'DATA' section contains all the information about manufacturing tasks and geometries. In this section, the project entity is the starting point of the program execution, which indicates the workplan to be executed and the workpiece upon which operations have to be performed. The workplan contains a series of manufacturing tasks including information of workpiece. It is composed of three different types of executables: workingsteps, NC functions and program structures. Among those workingsteps, executable is the most important because it defines manufacturing features such as 2.5D and 3D regions. Each workingstep also defines sub-features such as planar_face, pocket, slot, round-hole, etc., with cutting condition information. The workingstep also contains features and machining operations data; the features of workingstep contain geometrical data and the machining operations contain generally omitted tool paths, tool and machining functions data including machining information and NC commands. In particular, this includes data regarding tools, machining strategies, definitions of workpiece, etc. For further details regarding the ISO STEP-NC file structure, refer Calabrese and Celentano (2007), Krzic, Stoic, and Kopac (2009), Y. Zhang et al. (2010) and Xu and He (2004).

The introduction of ISO 14649 recovers the information loss from CAD/CAPP/CAM to CNC and makes the CNC more open, interoperable and intelligent (Xu and Newman 2006). STEP-NC is mainly focused on exchange of data with the machine tool controller. In order to achieve all the aims of modern manufacturing systems, a major problem of CNC machine vendor's specifications dependence was faced. According to Mori et al. (2001), 'Most of the today's CNC machine tool systems are being equipped with CNC controller supplied by controller vendors as a "black box" and this makes it difficult for the machine tool builder to quickly develop and implement the new/custom control functions'. In order to overcome the shortcomings of vendor's dependency, an open architecture control (OAC) technology-based machine controllers need to be developed. Open controller means 'controllers independent from manufactures technology, allowing the user to buy hardware and software from several different manufacturers and freely assemble the acquired piece of equipment' (Asato et al. 2002). Zheng, Zhao, and Li (2005) state that 'Personal Computer (PC) has been one of the preferred hardware platform for open CNC system because of its good openness and high performance price ratio'. With the development of manufacturing world, Ma et al.(2007) highlight that the current trend of CNC machines is towards the development of personal computer (PC)-based Soft-CNC systems. The implementation of OAC on CNC systems enables some hardware reconfiguration, communication and advanced NC programming technology in the new-generation CNC systems (Park, Kim, and Cho 2006).

The development of CNC systems based on open architecture has been the hot topic of research. In the race towards the development of the new-generation CNC system, nowadays more concentration is on STEP-compliant systems. Various technologies and many research works have been carried out all around the world in phase of different projects such as project OPTIMAL, project STEP-NC (EP 29708), project Super Model and Project IMS 97006 (Step-Compliant Data Interface for Numeric Controls). All these projects are being carried out with the calibration of various developed countries like the US, Germany, Sweden, Switzerland, Italy, South Korea, Japan, etc., with well-known developed companies of hardware and software like Siemens, OSAI, Open Mind, Dassault, CADCAMation, Cubictek, STEP Tools, etc. (Krzic, Stoic, and Kopac 2009). Although various different approaches were introduced by various scholars based on STEP implementation on CNC, those approaches had introduced various techniques of STEP-NC interpretation. The implementation of STEP-NC was initiated by the STEP to G-code or vice versa translation methods. Various scholars had introduced different approaches for this. For example, Newman, Allen, and Rosso (2003) and Newman and Nassehi (2007) introduced a multi-agent-based approach to translate STEP into G-code for implementation on CNC based on JAVA platform; Shin, Suh, and Stroud (2007) introduced a method of translation of G-code into STEP-NC and vice versa for lathe machine only; H. Wang, Xu, and Des Tedford (2007) and Minhat et al. (2009) developed a prototype system with interpreter based on functional blocks techniques that translate STEP-NC into the G-code for implementation; and X. Zhang et al. (2013) introduced a knowledge-based system that translates the G-code into STEP-NC for milling machine by utilising feature reorganisation technique on JAVA platform. These translation methods of STEP-NC implementation were unable to provide all the features of STEP-NC because of the data conversion from high level to low level (Xu and Newman 2006). In order to enable all the features of STEP-NC, direct interpretation is required. There were some techniques introduced by various scholars. For example, Lee et al. (2006) developed a prototype system with interpreter that converts STEP part 21 into part 28 format, the system provides data modification facilities and developed on VC++ platform; Kramer et al. (2006) introduced a real-time STEP-NC interpreter for FBICS software developed on C++ and ST-developer with rose library environments; Calabrese and Celentano (2007) developed a system developed on C language platform for both high- and low-level data models; and Pacheco et al. (2012) introduced a prototype system with interpreter for drilling process on JAVA platform. The major projects of STEP-NC are demonstrated in different companies and institutes at different regions of the world; the output of these demonstrations was very satisfactory and proved very beneficial for industries to have belief in STEP-NC as it had answered many of the questions from the manufacturing world (Hardwick et al. 2013; Hamilton, Hascoet, and Rauch 2014). Similar approaches were also presented by various scholars for the development of ISO 6983-based open architecture CNC system such as Li and Zhang (2010), T. Wang, Liu, and Wang (2010), Xiao et al. (2007), Pabolu and Srinivas (2010), Khanna et al. (2013), Ekkachai et al. (2009), Ramesh, Poo, and Intelligent (2009), Zhanbiao (2010), da Rocha, Diogne De Silva E Souza, and De Lima Tostes (2010), etc.

From this short survey, it has been observed that for the development of these kinds of systems many areas need to be focused upon; however, in this study, software module of the CNC controller, commonly known as interpreter, has been highlighted. In previous approaches, the majority of scholars had utilised C, C++, JAVA, VB, etc., platforms in system development. In this approach, a new method of ISO data model interpretation based on the Laboratory Virtual Instrument Engineering Workbench (LabVIEW) platform has been introduced. In previous approaches, almost none had utilised that platform for ISO data model interpretation, although some had introduced CNC motion control techniques based on that platform such as Weidong and Zhanbiao (2010), Zhanbiao (2010), da Rocha, Diogne De Silva E Souza, and De Lima Tostes (2010), but these does not contain any software model for ISO data interface model interpretation. So in this approach, a new interpretation module for ISO 14649 and 6983 based on the LabVIEW platform has been introduced. The overall design of the paper includes, firstly, an introduction, followed by structural design, algorithms design, graphical user interface (GUI), working principle, experimental study, and ends with some discussion, conclusion, and future works.

2. Structural design

This module is specially designed for PC based on OAC CNC systems. It is programmed in National Instrument (NI) LabVIEW with some special functions. The function of the developed module is to translate ISO 14649 and ISO 6983 code commands as per the internal structure of the CNC machine. The module interprets the acceleration, deceleration, spindle speed, feed rate, tool, tolerance, etc., data from ISO data interface model codes, so that the machine can be moved throughout linear or circular interpolation and perform operations. The LabVIEW platform was chosen because it is very popular and proved to be a very useful tool in control, display, analysis and data acquisition applications. LabVIEW contains a rich set of tool kits to provide a single platform for various activities. It is a graphical language easy to use compared to text coding platforms. LabVIEW features include libraries of reusable code, support for building GUIs, use of the dataflow paradigm and automatic memory management (Weidong and Zhanbiao 2010; Zhanbiao 2010). As ISO 14649 data model contains a rich set of information and ISO 6983 contains low-level information, LabVIEW can be a very useful platform to provide all in one unit for CNC by utilising this information. The major advantage of LabVIEW is that it provides a wide range of connectivity in hardware and software terms that makes an open environment for CNC. In hardware terms, LabVIEW is able to provide a wide range of connectivity through serial ports, wireless, etc. (LabVIEW 2009; Bishop 2009; Elliott et al. 2007). Whereas in the software terms it is able to communicate with various third-party software like MS Office, C, C++, C#, Java, MATLAB, SolidWorks, Visual Studio, etc. Another major advantage of LabVIEW is that its program can be directly applied to various tasks without having to change the program code and the parallelisation of the code execution can be achieved easily (Weidong and Zhanbiao 2010). These facilities of LabVIEW can enable access to monitoring, inspection, database, Internet connectivity, etc., functions at a single platform; in other words, it can be a useful source to make an all-in-one platform for the CNC machine. These facilities can make a LabVIEW platform very useful in the development of modern CNC systems. The developed module is able to provide shop floor editing, output file generation and easy data representation facilities into the CNC system. The module also provides a platform for a wide range of hardware and software connectivity. The architecture of the developed module is composed of three sub-modules – *information, abstraction* and *production* – which contains various functional blocks such as *case route, recite, guide route, exploration, match, excerpt, token, stock, chain, tool path generator, output* and *physical file maker* as shown in Figure 1.

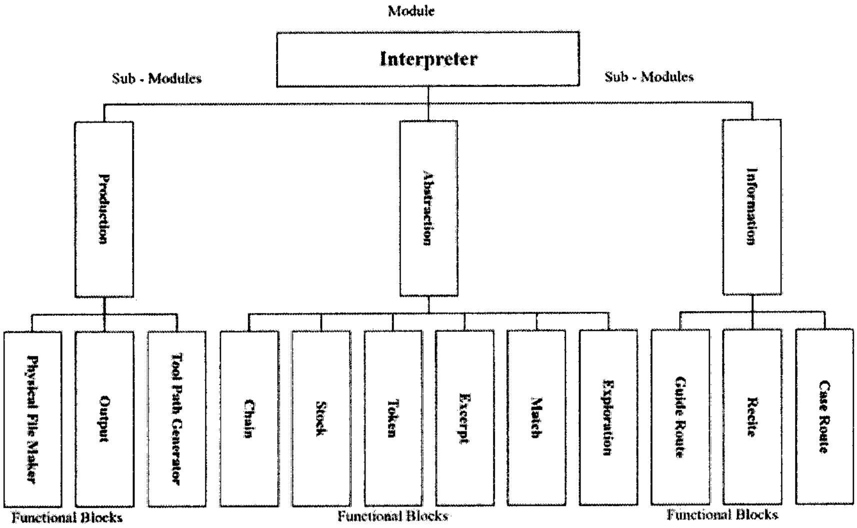


Figure 1. Structural design of the interpretation module.

2.1. *Information module*

This sub-module is the initial phase of the interpreter. The purpose of this module is to provide the path control for the file uploading, and in the meantime it also enables the *recite* function that reads all the contents of the input file. This module also contains the *guide route* function which guides the interpreter regarding the starting point of the input file. Overall, this module is composed of the *case route, recite* and *guide route* functions blocks. The functionality of this module is initiated by the generation of path control to get the input source code. After getting the source code path, the next step of the *recite* function is activated that reads the complete content of the input source code. At the end of this phase, the *guide route* function is executed. The role of this function is to guide the interpreter regarding the starting point of the code reading. The enabling of this function makes the interpreter to read the code from any line of the input file; by default this function has zero value that indicates the first line of the input code.

2.2. *Abstraction module*

This sub-module is the combination of both lexical scan and syntax analysis functions. These functions are achieved by various functional blocks such as *exploration, match, excerpt, token, stock* and *chain*. First of all, the *exploration* function of this module is activated, which searches for the patterns and entities in cases of ISO 6983 and ISO 14649, respectively. In the case of ISO 14649, the starting point is titled 'PROJECT' string; therefore, the module first searches for that string inside the 'DATA' section of the ISO 14649 code. After that, the function starts searching for the entity numbers (i.e. #2, #4) inside the input file. However, in the G-code case, the function searches for the pattern (X, Y, Z, T, S, F, etc.) inside the input code. After that, all the founded values (entity numbers or patters) go to the *match* function, which extracts the entire data of searched entity numbers or patterns from the input file. After that, the matched data goes to the *excerpt* function that mines the data as per the process type, that is, extract names, values, entity numbers, etc., in case of ISO 14649. The mining module has different sub-modules for different machining process because in the STEP-NC standard there are some differences in the placement of data for different process. However, in the case of ISO 6983, this function will extract the data values from the matched pattern. At the end of the *excerpt* function, the *token* function is executed that breaks all the extracted data into string tokens. The aim of this function is to separate the data as per their type in arrays such as for ISO 14649 (i.e. entity names in one array, values in other array, entity numbers in another array, etc.) and similarly for ISO 6983 (i.e. X pattern in one array, Y pattern in other, etc.). These generated arrays are then passed to the *stock* function, which stores all the information inside its memory. At the end of this module, the function of the *chain* takes place, which combines all the extracted token data (pattern values or values with entity names) into a single code and passes to the *production module*. Figure 2 shows the functionality of the *abstraction module*.

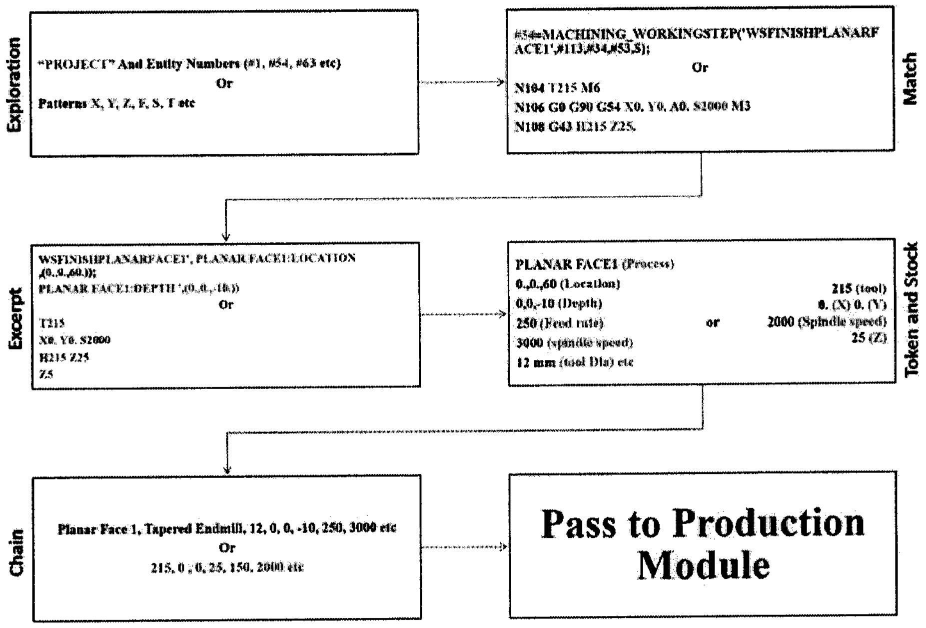


Figure 2. Functionality of abstraction module.

2.3. *Production module*

This sub-module is the last phase of the interpreter. It takes the output of the *abstraction* sub-module as an input. The initial function of this module is the *tool path generator* that takes the combined code of the *abstraction module* as an input and generates tool paths by utilising the values of the combined code. The *tool path generator* function contains *facing, drilling* and *pocketing* tool path generator sub-functional blocks. The activation of sub-function is recognised from the process names, which are included in the combined code of the *abstraction module*. After the *tool path generation*, the module passes the data to the *output* function that will transfer the data to the relative hardware parts of the CNC machine controller. The *output* function plays a key role in passing the data to the machine controller hardware parts in the correct way (i.e. axis values go to the respective *X, Y* and *Z* axis servo/stepper motors, tool values go to the tool changer, etc.). At the same time, this module is also able to generate physical output file in user-defined structure of .txt and .xml formats by means of the *physical file generator* function. The generated files can be used in any type of PC-based open architecture CNC machine systems.

3. Algorithm design

The developed module is composed of two algorithms because the ISO 6983 data interface model is a line-based language and can be executed line by line. Whereas the ISO 14649 is intended to be executed workingstep by workingstep. The STEP-NC algorithm must read the complete program first and able to generate tool paths (Kramer et al. 2006). These algorithms utilise various functional blocks for the interpretation of the ISO data interface modules.

3.1. *Algorithm design for ISO 14649 interpretation*

The algorithm of ISO 14649 interpretation starts with the *case route* function that provides the path to load ISO 14649 part 21 file. Then the *recite* function is executed, which reads the complete content of the input file. This function has a decision module that looks inside the input file and makes decisions regarding the status of file. If the input file does not contain any ISO 14649 data, the decision model sends the signal towards the error message. If it contains data, the model passes the file to the *guide route* function. The aim of this function is to provide access to read the input file from any point.

After the selection of line index, the data passes to the *exploration* function. This function first searches for the 'PROJECT' string inside the input code, followed by the search for entity numbers (#1, #2, etc.) inside the input file. While reading for the first time if the function has not found the 'PROJECT' string or entity number, it will pass the signals to error message. In other cases, the founded entity numbers are passed to the *match* function that reads the input file and searches for the entities number values. The output of this function is the complete data of the entity number (i.e. #54 = MACHINING_WORKINGSTEP ('WSFINISHPLANARFACE1', #113,#34,#53,$)), then that data is entered into the *excerpt* function that first searches for the machining process and then extracts the data as per process types. This function is linked with the process names library that guides the interpreter regarding the data mining structure. In the STEP-NC file, each process has some difference in a series of information placements because different processes contain different amounts of information. After that, the *token* function is executed that generates the string tokens of the extracted data. The generated string is stored in the internal memory of the system with the help of the *stock* function. The *token* function divides the data into separate arrays such as entity numbers array, process names array, locations values array, depth values array, tolerance values array, etc. At the end, the *token* function passes all the information to the *stock* function that saves all the data in the internal memory. After saving, the function recalls the *exploration* function to search for entity numbers inside all the stocked arrays. At this stage, the algorithm makes a decision. If the entity number is found, it recalls all the processes again. However, when no entity number is found in the stored data arrays, the function parses the data to the *chain* function that combines all the data into a single code and passes to the *production module.*

In this module, the combined code is passed to the *tool path generator* function that utilises all the information of the combined code and generates tool paths for specific processes. This function is also connected to the process name library that guides the function regarding the selection of sub-tool path generation modules as per specific process types (i.e. facing, drilling, pocket, etc.). After the *tool path generation*, the *output* function is executed that parses the information to the hardware parts of the CNC machine. At the same time, this module is able to generate the output physical file in .txt and .xml formats as per user-defined structure with the help of the *physical file maker* function. Figure 3 shows the flow chart of the ISO 14649 algorithm design, whereas Figure 4 shows the functionality of the algorithm.

3.2. *ISO 6983 interpretation algorithm design*

This algorithm is also composed of three sub-modules: *information, abstraction* and *production*. The functionality of the first and last modules is the same as discussed in the ISO 14649 interpret algorithm design. Whereas there is only a slight difference in the *abstraction module.* This module starts with the *exploration* function that searches

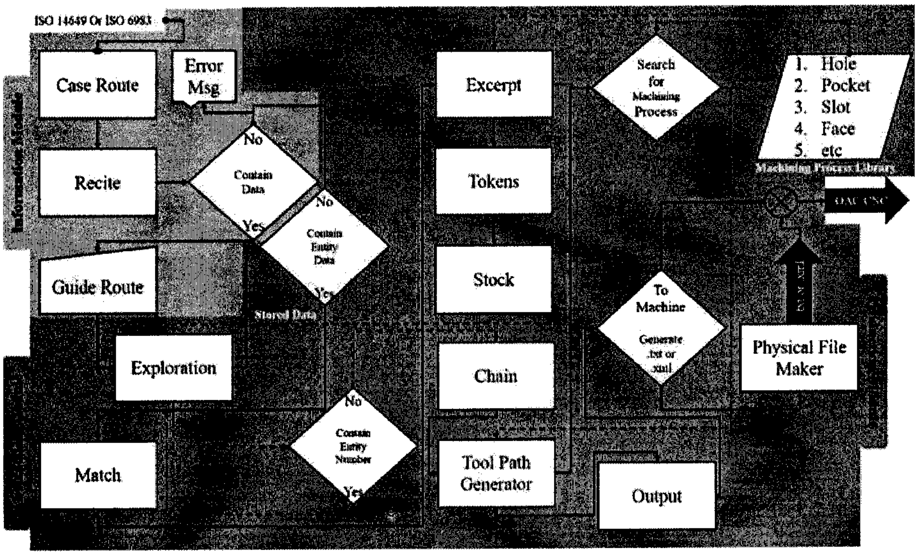
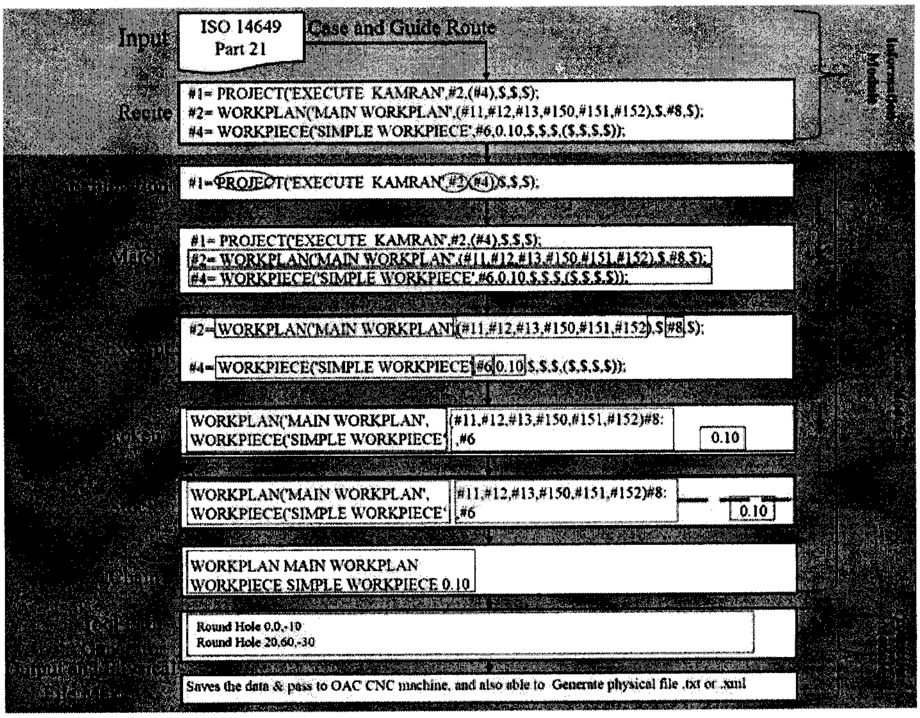


Figure 3. Flow chart of the ISO 14649 algorithm design.



```
#1= PROJECT('EXECUTE  KAMRAN',#2,(#4),$,$,$);
#2= WORKPLAN('MAIN WORKPLAN',(#11,#12,#13,#150,#151,#152),$,#8,$);
#4= WORKPIECE('SIMPLE WORKPIECE',#6,0.10,$,$,$,($,$,$,$));

#1=PROJECT('EXECUTE  KAMRAN',#2,(#4),$,$,$);

#1= PROJECT('EXECUTE  KAMRAN',#2,(#4),$,$,$);
#2= WORKPLAN('MAIN WORKPLAN',(#11,#12,#13,#150,#151,#152),$,#8,$);
#4= WORKPIECE('SIMPLE WORKPIECE',#6,0.10,$,$,$,($,$,$,$));

#2=WORKPLAN('MAIN WORKPLAN',(#11,#12,#13,#150,#151,#152),$,#8,$);

#4=WORKPIECE('SIMPLE WORKPIECE',#6,0.10,$,$,$,($,$,$,$));

WORKPLAN('MAIN WORKPLAN',      (#11,#12,#13,#150,#151,#152)#8:
WORKPIECE('SIMPLE WORKPIECE',  ,#6                               0.10

WORKPLAN('MAIN WORKPLAN',      #11,#12,#13,#150,#151,#152)#8:
WORKPIECE('SIMPLE WORKPIECE'   ,#6                               0.10

WORKPLAN MAIN WORKPLAN
WORKPIECE SIMPLE WORKPIECE 0.10

Round Hole 0,0,-10
Round Hole 20,60,-30

Saves the data & pass to OAC CNC machine, and also able to  Generate physical file .txt or .xml
```

Figure 4. Functionality of the ISO 14649 algorithm.

for patterns (i.e. X, Y, Z, F, S, T, etc.) inside the input code. This function has a decision module that passes the data to error message if no pattern is found. However, in other cases, the data passes to the *match, excerpt, token, stock* and *chain* functions step by step. The *match* function of the algorithm scans the input file for pattern values (i.e. X25, Y30, Z-5, etc.). These values will be extracted by *excerpt* function and parsed to the *token* function. This function breaks the extracted data into token strings in the form of arrays. This function breaks the data into different arrays such as *X* axis values in one array, *Y* axis values in other array, tool numbers in another array, etc. After that, the *token* function parses the data to the *stock* function. The aim of the *stock* function is to save all the data in the

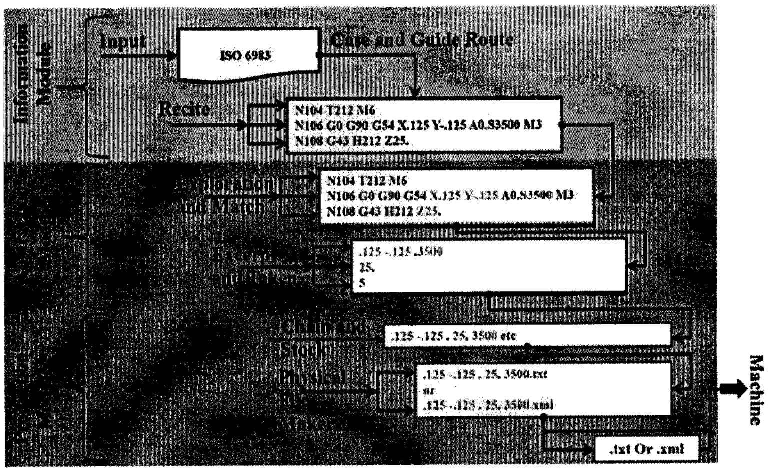


Figure 5. Functionality of the ISO 6983 algorithm.

internal memory of the system. Then that saved data is recited by the *chain* function that combines all the data into single code and parses the combined code to the *production module*. That combined code provides a shop floor editing functionalities for position, tool, spindle, feed rate, etc., data. The functionality of the ISO 6983 algorithm is shown in Figure 5.

4. Graphical user interfaces

The GUIs of a developed module is composed of four tabs: *main, shop floor editor, details* and *setting*. The *main* tab contains *case route* function from where the system gets the input code. The ISO 14649 GUI shows the complete contents of the input file with headers data, project name data, workpiece data, cutting tool data and machining process data. The main screen of GUI shows all the data of header, project name, workpiece, cutting tool and machining processes in different sections, so that a user can easily understand the content of the input code. Whereas the ISO 6983 GUI *main* page shows the input file path control, input ISO 6983 code and interpreted code. The *shop floor editor* tab shows the combined code with tool paths in separate fields for easy understanding and modifications in information offline or while machining. This tab shows all the information regarding positions, tool, spindle, feed rate, tolerance, etc., in separate columns. The *details* tab provides the complete information regarding position, tool, setup, workpiece, processes, tolerances, locations, depths, etc., in different sections with patterns or entity numbers, patterns or entity names, values, etc., and also provides an access for data modification offline or while machining. The last tab is of *settings* that provide access to the input and output setting of the file. For input setting, it has the *guide route* function, whereas for output setting it has a *physical file maker* function, where the user can set the structure of the output file as per the required structure of the CNC machine in .txt and .xml formats. The input setting is to guide the interpreter regarding the starting point (the point from which the interpreter starts reading the input code). By default this has a value of 0 that indicates the first line of the input code. Output settings have three steps: the first step is for selecting the output file format (.txt or .xml), the second step is for format and delimiter settings for text file format and the last step is for .xml output settings using XML encoding controller. Figure 6 shows the snapshots of the GUIs.

5. Working principle

The developed module is able to work with current CAD/CAM systems. In the case of ISO 14649, the working principle starts with uploading of the STEP part 21 file. There are few platforms available for the generation of STEP part 21 file. In this study, GEN-MILL generator (Yusof, Kassim, and Zamri Tan 2011), ST-Developer and STEP Tools online platforms (http://www.steptools.com) are used. Whereas in case of ISO 6983, the working cycle starts with the CAD design which then utilises CAM software for the generation of ISO 6983 code. After file uploading, the system performs the functions of the *information module* first, *abstraction module* second and *production module* third as shown in Figure 7.

6. Experimental study

In this study, various experiments have been performed for the testing and validation of the developed interpreter. Two of these are discussed in this section: one is based on the ISO 14649 interpretation and the other is based on the ISO 6983 interpretation.

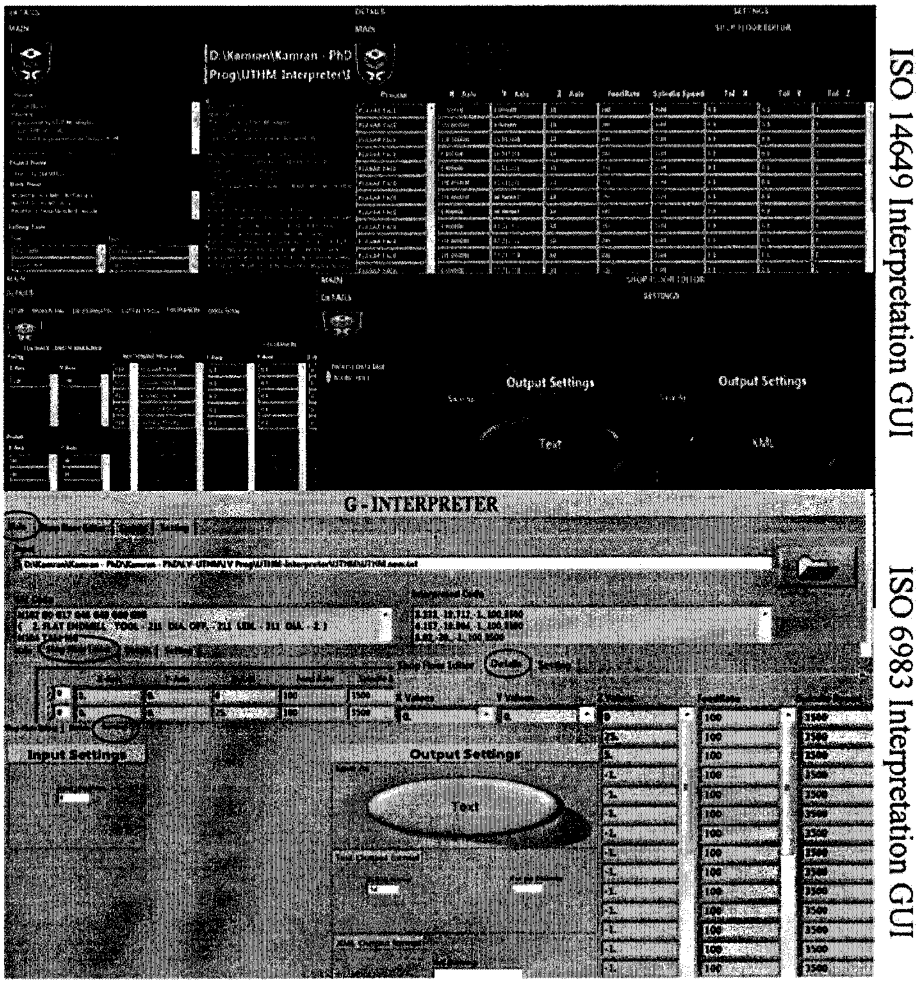


Figure 6. Snapshots of the developed module GUIs.

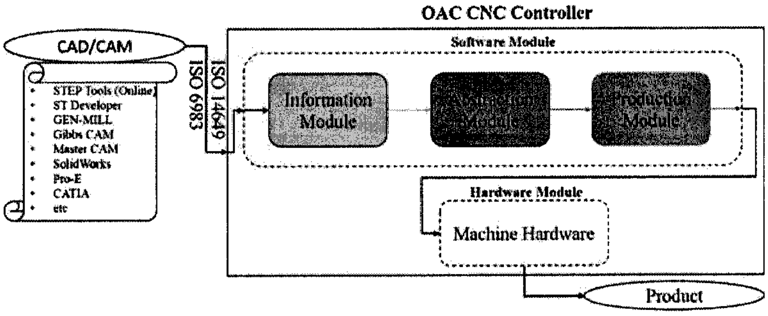


Figure 7. Working principle of the developed module.

6.1. *Experiment based on ISO 14649 interpretation*

In this study, the manufacturing of ISO 14649 example 1 part has been carried out by using the developed module with the OAC CNC system (Yusof and Latif 2013). The experiment started with the generation of STEP part 21 file by using STEP Tools online platform. After that, the file is uploaded into the developed interpretation module. Before performing the interpretation operation, the setting

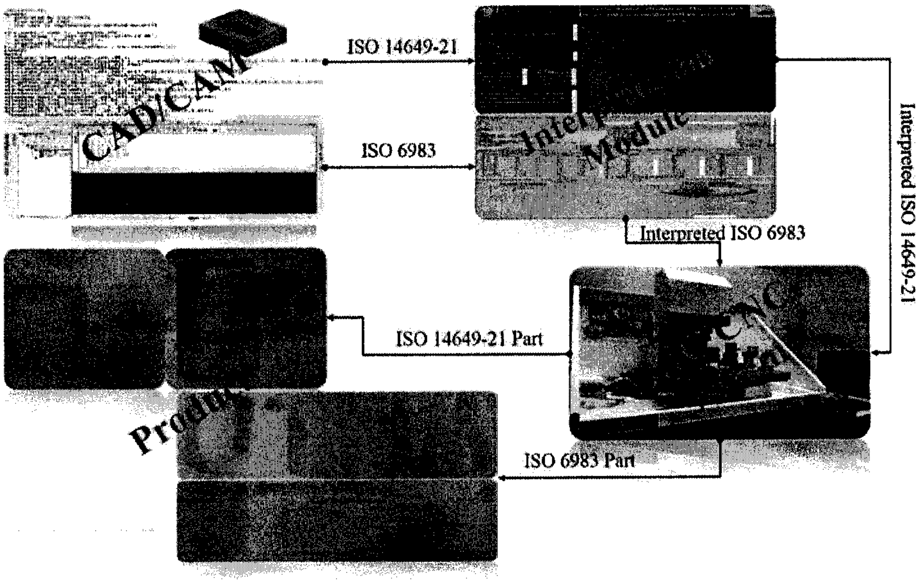


Figure 8. Experimental set-up of the developed module.

of the output file is carried out as per the requirements from the *setting* tab of the developed module. In this experiment, the output code is formatted as floating point number (%f) with comma (,) delimitation. After all settings have been done the example 1 part 21 file is executed by the developed interpreter and output physical file in .txt format is generated. Then the generated file is uploaded in the OAC CNC system via USB. The OAC system gets the information from the interpreted code and performed operations for the production of part, as shown in Figure 8.

6.2. Experiment based on ISO 6983 interpretation

This experiment is to write the letters U, T, H and M on a piece of wax material by using current CAD/CAM system, the developed interpreter and open architecture CNC machine (Yusof and Latif 2013). The experiment starts with the design process, in which U, T, H and M letters were designed by using CAD/CAM software. After designing, CAM machine features are given to the designer in order to generate the ISO 6983 code and the generated code is saved in .txt format by using the CAM software code editor. Once the ISO 6983 code is generated, the next step of uploading the code into the developed interpretation module was carried out. Before performing the interpretation operation, the setting of the output file is carried out as per the requirements from the *setting* tab of the developed module. In this experiment, the output code is formatted as floating point number (%f) with comma (,) delimitation. After all settings have been done, the code is executed and the output file saved in .txt format. Then, finally, the interpreted code is transferred to an OAC CNC machine and operations are performed as shown in Figure 8.

7. Discussion, conclusion and future works

In this paper, an interpretation module has been developed for PC-based open architecture CNC control system that is able to interpret the ISO 14649 part 21 and ISO 6983 data interface model information as per the required structure of the CNC machine. The module is composed of three sub-modules: *information, abstraction* and *production*. These modules works on the basis of the various functional blocks such as *case route, recite, guide route, exploration, match, excerpt, token, stock, chain, tool path generator, output* and *physical file maker*. The developed module also provides data editing facilities while working or offline and is also able to generate physical file output in .txt and .xml formats as per user-defined structure.

The module is designed in NI LabVIEW and is of open nature that welcomes contributions from different sources to communicate in terms of software (VB, C++, C#, JAVA, MATLAB, SolidWorks, etc.) and hardware to enable monitoring, inspection, report generation, etc., facilities. In the ISO 6983 case, these facilities are limited because of low-level data; however, ISO 14649 contains high-level data and can be a strong competitor to replace the G-code. The STEP-NC interpreter is easy to develop compared with the G-code because the G-code requires post-processors for specific machines. The STEP-NC interpreter can be easily implemented on PC-based CNC systems and a single set of algorithm can enable all the features of high-level data interface model and makes CNC more open, flexible, intelligent, etc.

This module can make life easy for the users. It extracts the complete data from high-level and low-level codes and shows an easy way on GUIs. It shows all the extracted data of set-up, workpiece, coordinates, locations, depth, tool, tolerance, etc., into separate portions to provide an easy understanding to the user about the content of the input file and also provides options for easy data modification as per the requirements. This system is of low cost compared to commercial systems and can play an important role in the development of small industries and shop floor CNC users. The major advantage of this system is that it can be used to convert old manually operated machines into automatic units. The system can be easily implemented on any system without any hardware or software requirements. Currently, the system has been implemented on Windows environment-based CNC system. This study also shows the validation of the developed interpretation module in experimental work. There are a number of successful experiments performed with the developed module that shows good accuracy and performance with the OAC CNC system.

Overall, this study contributes in the field of interpreter by introducing a new method of ISO data interface models interpretation and utilisation of LabVIEW platform in the development of STEP-compliant system. The system provides a platform to enable modern manufacturing facilities like shop floor editing, inspection, monitoring, report generation, etc., in current and future manufacturing environment.

In future, the interpreter can be updated by enabling universal interpretation for ISO 6983 like Guo et al. (2008, 2012), G-code to STEP-NC or vice versa translation like Shin, Suh, and Stroud (2007) and X. Zhang et al. (2013), implement on Linux CNC environment to enable RT environment like Ji, Li, and Wang (2008) and Staroveški et al. (2009), ISO 14649 part 28 interpretation, intelligent and accurate tool path generations, monitoring and inspection GUIs development, simulation, more processes interpretation and tool path generation, etc. Overall, the main aim of this research is to develop an all-in-one platform for the CNC machine.

Acknowledgement

The authors thank Dr. David Loffredo, Vice President of STEP Tools Inc., and Dr. Nazareno de Oliveira Pacheco from the Universidade Federal Santa Catarina for their help and guidance.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Malaysian Government under Science fund S021 (MOSTI) and Malaysian International Scholarship (MIS) under the Ministry of Education (MOE) Malaysia.

References

Asato, O., E. Kato, R. Inamasu, and A. Porto. 2002. "Analysis of Open CNC Architecture for Machine Tools." *Journal of the Brazilian Society of Mechanical Sciences* 24 (3): 208–212. doi:10.1590/S0100-73862002000300009.

Bishop, R. H. 2009. *LabVIEW 2009 Student Edition*. Upper Saddle River, NJ: Prentice Hall Press.

Calabrese, F., and G. Celentano. 2007. "Design and Realization of a STEP-NC Compliant CNC Embedded Controller." IEEE Conference on Emerging Technologies and Factory Automation, 2007. ETFA, Patras, September 25–28, 1010–1017. IEEE.

da Rocha, P., R. Diogne De Silva E Souza, and M. E. De Lima Tostes. 2010. "Prototype CNC Machine Design." 2010 9th IEEE/IAS International Conference on Industry Applications (INDUSCON), Sao Paulo, November 8–10, 1–5. IEEE.

Ekkachai, K., U. Komin, W. Chaopramualkul, A. Tantaworrasilp, P. Kwansud, P. Seekhao, T. Leelasawassuk, K. Tanta-Ngai, and K. Tungpimolrut. 2009. "Design and Development of an Open Architecture CNC Controller for Milling Machine Retrofitting." ICCAS-SICE, Fukuoka, August 18–21, 5629–5632. IEEE.

Elliott, C., V. Vijayakumar, W. Zink, and R. Hansen. 2007. "National Instruments Labview: A Programming Environment for Laboratory Automation and Measurement." *Journal of the Association for Laboratory Automation* 12 (1): 17–24. doi:10.1016/j.jala.2006.07.012.

Ertell, G. G. 1969. *Numerical Control*. New York: Wiley.

Feeney, A. B. 2002. "The STEP Modular Architecture." *Journal of Computing and Information Science in Engineering* 2 (2): 132–135. doi:10.1115/1.1511520.

Feeney, A. B., T. Kramer, F. Proctor, M. Hardwick, and D. Loffredo. 2003. "STEP-NC Implementation—Arm or AIM?" ISO T24 STEP-Manufacturing Meeting, San Diego, CA, March. ASME.

Groover, M. P. 2007. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Upper Saddle River, NJ: Prentice Hall Press.

Guo, X., Y. Liu, D. Du, K. Yamazaki, and M. Fujishima. 2012. "A Universal NC Program Processor Design and Prototype Implementation for CNC Systems." *The International Journal of Advanced Manufacturing Technology* 60 (5–8): 561–575. doi:10.1007/s00170-011-3618-6.

Guo, X., Y. Liu, K. Yamazaki, K. Kashihara, and M. Fujishima. 2008. "A Study of A Universal NC Program Processor for A CNC System." *The International Journal of Advanced Manufacturing Technology* 36 (7–8): 738–745. doi:10.1007/s00170-006-0890-y.

Hamilton, K., J.-Y. Hascoet, and M. Rauch. 2014. "Implementing STEP-NC: Exploring Possibilities for the Future of Advanced Manufacturing." In *Modern Mechanical Engineering*, 199–239. Berlin: Springer.

Hardwick, M., Y. F. Zhao, F. M. Proctor, A. Nassehi, X. Xu, S. Venkatesh, D. Odendahl, L. Xu, M. Hedlind, M. Lundgren, L. Maggiano, D. Loffredo, J. Fritz, B. Olsson, J. Garrido, and A. Brail. 2013. "A Roadmap for STEP-NC-Enabled Interoperable Manufacturing." *The International Journal of*

Advanced Manufacturing Technology 68 (5–8): 1023–1037. doi:10.1007/s00170-013-4894-0.

Houshmand, M., and O. F. Valilai. 2012. "LAYMOD: A Layered and Modular Platform for Cax Product Data Integration Based on the Modular Architecture of the Standard for Exchange of Product Data." *International Journal of Computer Integrated Manufacturing* 25 (6): 473–487. doi:10.1080/0951192X.2011.646308.

ISO 6983-1. 1982. *Numerical Control of Machines-Program Format and Definition of Address Words-Part 1: Data Format for Positioning, Line and Contouring Control Systems*. Geneva: ISO.

ISO, C. 1991. "10303-1: Product Data Representation and Exchange-Part 1: Overview and Fundamental Principles." In *TC*, 184. Geneva: ISO.

ISO, I. 1994. *10303-1. Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part, 1*. Geneva: ISO.

ISO, T. 1994. *184/SC 4, ISO 10303-11: 1994 Industrial Automation Systems and Integration-Product Data Representation and Exchange-Part 11: Description Methods: The EXPRESS Language Reference Manual*. Geneva: ISO.

Ji, H., Y. Li, and J. Wang. 2008. "A Software Oriented CNC System Based on Linux/Rtlinux." *The International Journal of Advanced Manufacturing Technology* 39 (3–4): 291–301. doi:10.1007/s00170-007-1216-4.

Khanna, A., A. Kumar, A. Bhatnagar, R. Tyagi, and S. Srivastava. 2013. "Low-Cost Production CNC System." 2013 7th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, January 4–5, 523–528. IEEE.

Kramer, T. R., F. Proctor, X. Xu, and J. Michaloski. 2006. "Run-Time Interpretation of STEP-NC: Implementation and Performance." *International Journal of Computer Integrated Manufacturing* 19 (6): 495–507. doi:10.1080/09511920600622056.

Krzic, P., A. Stoic, and J. Kopac. 2009. "STEP-NC: A New Programming Code for the CNC Machines." *Strojniški Vestnik* 55 (6): 406–417.

LabVIEW, F. 2009. *National Instruments LabVIEW FPGA*. Austin, TX: Austin Press.

Lee, W., Y.-B. Bang, M. Ryou, W. Kwon, and H. Jee. 2006. "Development of a PC-Based Milling Machine Operated by STEP-NC in XML Format." *International Journal of Computer Integrated Manufacturing* 19 (6): 593–602. doi:10.1080/09511920600623674.

Li, M.-S., and F.-F. Zhang. 2010. "The Research of Embedded CNC System Component." 2010 International Conference on Computer Application and System Modeling (ICCASM), Vol. 7, North University of China, Taiyuan, October 22–24, V7-601–V607-604. IEEE.

Liang, S. Y., R. L. Hecker, and R. G. Landers. 2004. "Machining Process Monitoring and Control: The State-Of-The-Art." *Journal of Manufacturing Science and Engineering* 126 (2): 297–310. doi:10.1115/1.1707035.

Ma, X.-B., Z.-Y. Han, Y.-Z. Wang, and H.-Y. Fu. 2007. "Development of a PC-Based Open Architecture Software-CNC System." *Chinese Journal of Aeronautics* 20 (3): 272–281. doi:10.1016/S1000-9361(07)60044-2.

Minhat, M., V. Vyatkin, X. Xu, S. Wong, and Z. Al-Bayaa. 2009. "A Novel Open CNC Architecture Based on STEP-NC Data Model and IEC 61499 Function Blocks." *Robotics and Computer-Integrated Manufacturing* 25 (3): 560–569. doi:10.1016/j.rcim.2008.03.021.

Mori, M., K. Yamazaki, M. Fujishima, J. Liu, and N. Furukawa. 2001. "A Study on Development of an Open Servo System for Intelligent Control of A CNC Machine Tool." *CIRP Annals-Manufacturing Technology* 50 (1): 247–250. doi:10.1016/S0007-8506(07)62115-5.

Newman, S., R. Allen, and R. Rosso Jr. 2003. "CAD/CAM Solutions for STEP-Compliant CNC Manufacture." *International Journal of Computer Integrated Manufacturing* 16 (7–8): 590–597. doi:10.1080/0951192031000115688.

Newman, S., and A. Nassehi. 2007. "Universal Manufacturing Platform for CNC Machining." *CIRP Annals-Manufacturing Technology* 56 (1): 459–462. doi:10.1016/j.cirp.2007.05.110.

Pabolu, V. K., and S. Srinivas. 2010. "Design and Implementation of a Three Dimensional CNC Machine." *International Journal on Computer Science and Engineering* 2 (8): 2567–2570.

Pacheco, N. D. O., E. Harbs, R. S. Rosso Jr, M. D. S. Hounsell, and J. C. E. Ferreira. 2012. "Application of The Step-NC Standard in a Computer Numerical Controlled Machining Device." *ABCM Symposium Series in Mechatronics* 5: 713–723.

Park, S., S.-H. Kim, and H. Cho. 2006. "Kernel Software for Efficiently Building, Re-Configuring, and Distributing an Open CNC Controller." *The International Journal of Advanced Manufacturing Technology* 27 (7–8): 788–796. doi:10.1007/s00170-004-2243-z.

Ramesh, R., S. Jyothirmai, and K. Lavanya. 2013. "Intelligent Automation of Design and Manufacturing in Machine Tools Using an Open Architecture Motion Controller." *Journal of Manufacturing Systems* 32 (1): 248–259. doi:10.1016/j.jmsy.2012.11.004.

Ramesh, R., A. Poo, and Intelligent. 2009. "Ethernet Based Open Architecture Control System for Machine Tools." IEEE International Conference on Intelligent Computing and Intelligent Systems, Vol. 2, Shanghai, November 20–22, 612–616. IEEE.

Safaieh, M., A. Nassehi, and S. T. Newman. 2013. "A Novel Methodology for Cross-Technology Interoperability in CNC Machining." *Robotics and Computer-Integrated Manufacturing* 29 (3): 79–87. doi:10.1016/j.rcim.2012.04.016.

Shin, S.-J., S.-H. Suh, and I. Stroud. 2007. "Reincarnation of G-Code Based Part Programs into STEP-NC for Turning Applications." *Computer-Aided Design* 39 (1): 1–16. doi:10.1016/j.cad.2006.08.005.

Staroveški, T., D. Brezak, T. Udiljak, and D. Majetić. 2009. "Implementation of a Linux-Based CNC Open Control System." International Conference on Production Engineering, Biograd, June 17–20.

Stenerson, J. S., and K. Curran. 2005. *Computer Numerical Control: Operation and Programming*. Upper Saddle River, NJ: Prentice-Hall.

Suh, S.-H., and S.-U. Cheon. 2002. "A Framework for an Intelligent CNC and Data Model." *The International Journal of Advanced Manufacturing Technology* 19 (10): 727–735. doi:10.1007/s001700200083.

Suh, S.-H., J.-H. Cho, and H.-D. Hong. 2002. "On the Architecture of Intelligent STEP-Compliant CNC." *International Journal of Computer Integrated Manufacturing* 15 (2): 168–177. doi:10.1080/09511920110056541.

Talavage, J. 1987. *Flexible Manufacturing Systems in Practice: Design: Analysis and Simulation*. Vol. 26. Boca Raton, FL: CRC Press.

Vichare, P., A. Nassehi, S. Kumar, and S. T. Newman. 2009. "A Unified Manufacturing Resource Model for Representing CNC Machining Systems." *Robotics and Computer-*

Integrated Manufacturing 25 (6): 999–1007. doi:10.1016/j.rcim.2009.04.014.

Wang, H., X. Xu, and J. Des Tedford. 2007. "An Adaptable CNC System Based on STEP-NC and Function Blocks." *International Journal of Production Research* 45 (17): 3809–3829. doi:10.1080/00207540600774075.

Wang, T., Q. Liu, and L. Wang. 2010. "An RTOS-Based Embedded CNC System." 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE), TBD, Vol. 2, Changchun, August 24–26, 33–36. IEEE.

Weidong, Y., and G. Zhanbiao. 2010. "An Open CNC Controller Based on Labview Software." 2010 International Conference on Computer Application and System Modeling (ICCASM), North University of China, Vol. 4, Taiyuan, October 22–24, V4-476–V474-479. IEEE.

Xiao, S., D. Li, Y. Lai, J. Wan, and S. Feng. 2007. "An Open Architecture Numerical Control System Based on Windows CE." IEEE International Conference on Control and Automation, 2007, Guangzhou, May 30–June 1, 1237–1240. IEEE.

Xu, X., and Q. He. 2004. "Striving for a Total Integration of CAD, CAPP, CAM and CNC." *Robotics and Computer-Integrated Manufacturing* 20 (2): 101–109. doi:10.1016/j.rcim.2003.08.003.

Xu, X., and S. Newman. 2006. "Making CNC Machine Tools More Open, Interoperable and Intelligent—A Review of the Technologies." *Computers in Industry* 57 (2): 141–152. doi:10.1016/j.compind.2005.06.002.

Yusof, Y., N. D. Kassim, and N. Z. Zamri Tan. 2011. "The Development of a New STEP-NC Code Generator (GEN-MILL)." *International Journal of Computer Integrated Manufacturing* 24 (2): 126–134. doi:10.1080/0951192X.2010.531289.

Yusof, Y., and K. Latif. 2013. "Frame Work of LV-UTHM: An ISO 14649 Based Open Control System for CNC Milling Machine." *Applied Mechanics and Materials* 330: 619–623. doi:10.4028/www.scientific.net/AMM.330.619.

Zhanbiao, G. 2010. "An Open CNC Controller Based on Labview Software." In *2010 International Conference on Computer Application and System Modeling (ICCASM* 2010), Vol. 4, North University of China, Taiyuan, October 22–24.

Zhang, X., A. Nassehi, M. Safaieh, and S. T. Newman. 2013. Process Comprehension for Shopfloor Manufacturing Knowledge Reuse. *International Journal of Production Research* 51 (23–24): 1–15.

Zhang, Y., X. Xu, X. Bai, and Y. Liu 2010. "Understanding the STEP-NC Data Model for Computer Numerical Control." 2010 2nd International Conference on Advanced Computer Control (ICACC), Vol. 1, Shenyand, March 27–29, 300–304. IEEE.

Zheng, J., W. Zhao, and Z. Li. 2005. "Open EDM CNC System Based on RT Linux." *Computer Integrated Manufacturing Systems-Beijing* 11 (8): 1179.