Available online at www.sciencedirect.com
ScienceDirect
Computer-Aided Design 39 (2007) 1–16
COMPUTER-AIDED DESIGN
www.elsevier.com/locate/cad

Reincarnation of G-code based part programs into STEP-NC for turning applications

Seung-Jun Shin[a,1], Suk-Hwan Suh[b,*], Ian Stroud[c,2]

[a] National Research Lab for STEP-NC/e-Manufacturing, Division of Mechanical Engineering for Emerging Technology, POSTECH, South Korea
[b] Center for Ubiquitous Manufacturing, POSTECH, San 31, Hyoja-dong, Pohang 790-784, South Korea
[c] CAD/CAM Laboratory of Ecole Polytechnique Federale de Lausanne, EPFL, STI-IPR-LICP, Batiment ME Station 9, CH-1015 Lausanne, Switzerland

Received 21 May 2006; accepted 24 August 2006

Abstract

As STEP-NC emerges as the new CNC control method and a fundamental means for realizing e-manufacturing, old manufacturing information based on the conventional manufacturing standard will become obsolete. In practice, replacement of G-code based part programs into STEP-NC is a huge task. In this paper, methods to interpret G-code based part programs into STEP-NC code are investigated. G-code is a compact, coded set of numbers for axis movements, while STEP-NC is very comprehensive and includes information about features, operations, strategies, cutting tools, and so on. It is thus very challenging to derive such comprehensive information from the low level G-code information. In this paper, we first clarify what should be given and what may be given, and then present algorithms for deriving STEP-NC information, such as geometric features, operations, etc., from the tool movement (G-code) based on expert reasoning. The algorithms are developed for the turning application. The developed algorithms were implemented and tested on G-code part programs used in actual practice.

Keywords: STEP-NC; G-code; e-Manufacturing; STEP-Manufacturing; Process plan; Turning machine; Legacy control

1. Introduction

1.1. Background

With the rapid advancement of information technology associated with manufacturing technology, the manufacturing environment has changed significantly since the last decade. High speed machining, high precision machining, and multi-axis machining have extensively enhanced the productivity and quality of manufacturing. Furthermore, advanced internet technology introduced a new paradigm of e-manufacturing, in which the so-called DABA (Design-Anywhere–Build-Anywhere) system can be realized via collaborative schemes in a distributed manufacturing system.

The key to the success of the e-manufacturing paradigm is a seamless data flow in the CAD–CAM–CNC chain. The STEP (ISO 10303 : STandard for the Exchange of the Product data) developed since the 1980s is the data standard for exchanging product data between CAD, CAPP and CAM systems. For a seamless data flow between CAM and CNC, STEP-NC (formalized as ISO 14649) has been newly established as an international standard. Compared with the old standard ISO 6983 (so called M&G-codes) currently used as the interface between CAM and CNC, STEP-NC includes a much richer information set including ‘what-to-make’ (geometry) and ‘how-to-make’ (process plan). Also, STEP-NC has been harmonized with the STEP set of standards as ISO 10303 AP238, which offers the possibility of seamless integration of application throughout design and manufacturing. Recently, STEP-Manufacturing – the pursuit of STEP in, STEP out, STEP throughout in the entire manufacturing process – has been given attention and ISO TC184 SC4 and SC1, the international specialist groups of product and production data, recommended such information-based manufacturing with the suite of STEP-Manufacturing [1]. The suite shows the functions and associated information standards in terms of STEP

* Corresponding author. Tel.: +82 54 279 2196.
E-mail addresses: head@postech.ac.kr (S.-J. Shin), shs@postech.ac.kr (S.-H. Suh), ian.stroud@epfl.ch (I. Stroud).
[1] Tel.: +82 54 279 8239.
[2] Tel.: +41 21 693 2949.

0010-4485/$ - see front matter
doi:10.1016/j.cad.2006.08.005

Table 1
STEP-Manufacturing Roadmap [9]

		1 step (the beginning period)	2 step (the employment period)	3 step (the completion period)
Objective/Benefit		STEP-Mfg infra introduction through the minimum investment	Merit acquisition of STEP-Mfg by STEP-Mfg settlement	e-Mfg paradigm implementation based on STEP-Mfg
Time frame		2 year (TBA)	3–4 year (TBA)	After 5 year (TBA)
Infra range		Intranet (in company)	Internet (in local area)	Internet (international)
Information exchange level		Hybrid (STEP, STEP-NC, G-code)	Partial STEP-Mfg (STEP AP203, ISO14649)	Full STEP-Mfg (STEP APs, ISO14649, . . .)
Implementation level	CNC	Type 1 (conventional control) via post-processing	Type 2 (new control) via new & w/STEP-NC interpreter (Siemens)	Type 3 (intelligent control) via new and & intelligent controller (TurnSTEP-ACS)
	CAD/CAM	Legacy software with STEP-NC interface (ST-Plan, ST-Machine)	STEP & STEP-NC based CAPP/CAM (PosSFP, TurnSTEP-CGS)	CAPP/CAM for intelligent STEP-Mfg (TurnSTEP-CES)
Required technology	STEP-x interface	STEP-NC interpreter Post-processor for Type 1 (STEP-NC→G-code)	STEP, STEP-NC interpreter **STEP-NC converter (G-code→STEP-NC)**	STEP, STEP-NC interpreter
	Web service	Web-service build-up in server side (settlement of web-service range)	STEP-Mfg application build-up in client side	Client–server harmonization and improvement
	DB build up	Local DB	Global DB (STEP-Mfg repository)	Global DB (STEP-Mfg repository)
Charge of participant (Role division)	Company	Intranet infra in company, STEP-Mfg introduction	STEP-Mfg infra employment	e-Mfg infra employment
	R&D center	STEP-Mfg component technology research and spread	Component technology development, conformance verification	Verification of reliability, conformance, interoperability
	Government	STEP-Mfg introduction support, local IT infra build-up business	Infra technology employment business, local IT infra build-up business	Commercial use business, certification business, IT infra enlargement (nation)

APs (Application Protocol) from product design to finished part.

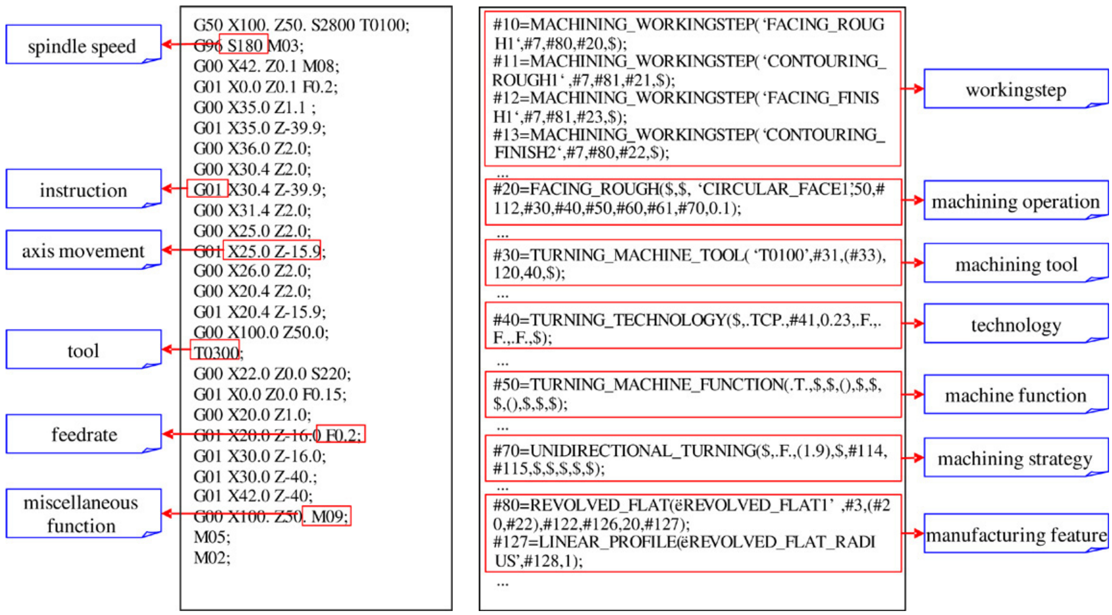
The impact of the new interface schemes can be visualized in many ways. As the new data models will work as an information highway for e-manufacturing encompassing CAD, CAM and CNC, the ‘art-to-part’ dream [2] can be realized, thereby producing a 3D model as a physical part by CNC, in the same way that a printer produces hard copy. Significant gains are expected in the process chain of CAD, CAM and CNC. Furthermore, complete elimination of post-processing is possible [2]. In the very near future, the new interface schemes will be used as a means for implementing internet B2B activities, e-design and e-manufacturing, as reported in several articles [3–8].

1.2. STEP-Manufacturing roadmap

Research and development on STEP-Manufacturing has been actively pursued and it has been demonstrated to work in practice both internationally and locally. At present, an effort has been made to apply the techniques to real industrial areas. However, truly, it is hard to realize full STEP-Manufacturing in one step due to the time, cost and technological difficulties. For this reason, Suh and Lee [9] suggested the STEP-Manufacturing Roadmap (Table 1) composed of three steps as the specific approach methodology for the formalization of the STEP-Manufacturing environment. The roadmap takes into consideration the following itemized strategies:

- Collaborative participation with many manufacturing-related companies
  - Collaborative interaction with design–engineering–machining company chains, CAD/CAM software users, CNC controller developers, CNC machine tool users and/or builders
- Inducement toward an information-oriented and international environment
  - Spread to information-oriented company and information exchange among collaborative companies
  - Gradual extension from local cluster to global environment and from metal working to other industrial sectors
- Consideration of compact and economical research and development
  - Practical use from conventional products to new intelligent STEP-based products
  - Inducement of implementation from partial to whole
  - Technology and service offered through Web service
  - Maximum utilization of accumulated know-how from R&D organizations

In the second step, STEP-NC is established as the international standard. STEP-NC part programs are the necessary input for the second type of STEP-CNC [10]. As a result, a great many conventional G-code part programs



(a) G-code part program. (b) STEP-NC part program.

Fig. 1. Comparison of G-code and STEP-NC formats.

will become obsolete. However, in reality, these G-code programs are very valuable because they incorporate both an implicit micro process plan as well as many years of operator experience. For this reason, a function for interpreting G-code into STEP-NC code is necessary in order to reincarnate old information.

The aim of this paper is to present a system for constructing STEP-NC from G-code as shown in bold in Table 1. This paper can contribute that: (1) time and cost can be reduced by automating conversion of G-code into STEP-NC. This is needed to overcome the problem of legacy code preserved by companies when STEP-CNC becomes widely available. (2) The early adoption of STEP-Manufacturing environment can be realized because a STEP-NC code can be generated easily by operators skilled in using G-code. (3) Accumulation of know-how from actual machining can be included due to storing in object-oriented structure.

In this paper, in Section 2, we first clarify what should be given and what may be given as inputs to the process, followed by design considerations and the functional architecture for G2STEP (software for interpreting G-code into STEP-NC) system in Section 3. Detailed descriptions of major functions needed for the system for the turning application are presented in Section 4. The prototype system has been developed and verified and the algorithms are presented in Section 5, followed by concluding remarks in Section 6.

2. Problem clarification

This section explains the different elements in the two control formats and the matching issue. A summary is given in Fig. 1. Also, this presents what should be given and what may be given.

2.1. G-code to be converted

As shown in Fig. 1(a), the current CNC language, so called M&G-codes which have been used since the 1950s, is a low-level language. This is defined by just numerical codes such as G, T, M, F, S indicating the movement of a machine and an axis to the controller. Since this delivers only limited information to the CNC (excluding valuable information such as part geometry and the process plan used to generate the NC code), it makes the CNC nothing but an executing mechanism completely unaware of the motions being executed. Also, it has become necessary to perform so called post-processing in order to convert the neutral part program for each controller.

2.2. STEP-NC code to be generated

STEP-NC is a new model for data transfer in the CAD–CAM–CNC chain. It remedies the shortcomings of G-code by specifying machining processes rather than the machine tool motion, using the object-oriented concept of workingsteps. The workingsteps represent the essential building blocks of manufacturing tasks. Each workingstep describes usually a single manufacturing operation using one tool and one strategy [11]. This has a manufacturing feature and a machining operation as shown in Fig. 1(b). Machining tool, machining strategy, technology and machine functions are included in machining operations. It is important to note that the tool path specification in ISO 14649 is 'optional', unlike in the current G-code where the tool path in terms of machine axes is the main information content.

2.3. Matching the control elements

This paper describes the conversion of lower level G-code, such as that shown in Fig. 1(a), into higher level STEP-NC,

Table 2
Definition and assumption of input condition

Type	Information	Description
Mandatory	G-code part program	No faulty and inefficient part program written by a skilled operator
		Application for one setup (no exchange of setup coordination system, when machining workpiece material in one spindle)
Mandatory	Tool information	Tool maker's catalogue model
		Holder and insert information represented by tool maker
		Compensation value for each launched tool
Mandatory	Instruction schema	Controller type
		M&G code instruction schema provided by controller maker
Optional	Drawing	CAD file for finished shape
Optional	Workpiece information	Geometry and material for workpiece

as shown in Fig. 1(b). In the case of the reverse, that is, from Fig. 1(b) to Fig. 1(a), R&D has advanced sufficiently, as described for various similar cases in [12–20] and is easier because conversion of tool movement from rich information is possible. Until now, though, research related to the inverse direction, the theme of this paper, has not been reported. There is only the APT2STEP application [21] in ST-Developer version 11 developed by STEPTools, Inc. This gives a solution for converting APT CL (Cutter Location) data into ISO 10303 AP238 CC1 (Conformance Class 1) [22] data, but this application offers the format transformation to ISO 10303 AP238.

2.4. Input condition clarification

As a matter of fact, there is no guarantee of success for conversion with only the G-code part program as input data. This is because there is no information about which tool is used, about what each code means and about the shape of an intended part. Hence we need to define what should be given, what may be given with other information – tool information, drawing, workpiece information, instruction schema of a controller – needed to execute a G-code part program on a machine. We cannot find out which operation is executed with only T-codes without additional detailed information in Fig. 1(a). Also, we need an instruction schema for the targeted controller, because each code is dependent on the controller type. For example, G70s and G80s mean cycle codes, G98 orders feed per minute in the FANUC0 series. However, G60s and G80s mean cycle codes, G94 instructs feed per minute in the FAGOR 8055T series. In contrast, drawing and workpiece information may be needed for feature recognition, but features can be inferred from technological analysis with tool movement.

As the result of consideration about the combination of input conditions, we can define three pieces of information – G-code part program, tool information, instruction schema – as the minimum mandatory inputs and two pieces of information – drawing, workpiece information – as the optional inputs. Also, this combination implies greater difficulty than if the whole information set is considered as mandatory. More detailed definitions and assumptions of input conditions are summarized in Table 2.

2.5. Notes for interpreting methodology and feature recognition

G-code consists of a monolithic, unstructured block of information whereas STEP-NC is divided into logical units. As an initial step, then, it is necessary to impose a structure on the G-code. There are a number of cues which can be used to subdivide G-code into logical blocks. Each logical block corresponds roughly to a feature. Feature recognition is normally done on the basis of a geometric shape and this information is one part of process planning. The features that can be inferred from the G-code are a mixture of geometry and other technological decisions. For turning, for example, the shape of the exterior of the part needs to be broken into parts so that it can be gripped. For milling the same process of G-code segmentation is required but there are different cues for the subdivision. The feature then has to be subdivided into groups which correspond to workingsteps in STEP-NC.

3. G2STEP architecture

This section describes the architecture of the G2STEP system. Section 3.1 presents the design considerations and Section 3.2 the functional architecture.

3.1. Design considerations for the G2STEP system

The G2STEP system generates a STEP-NC part program from a G-code part program with additional information related to real machining. The final objective of this system is that the final shape, machined using the STEP-NC, corresponds completely to that machined by G-code. In designing the details of the G2STEP system, various aspects, especially the following items, should be taken into consideration when designing the architecture of G2STEP for turning and milling.

- Restoration of real examples with a minimum of information
  It is common that the custody and the maintenance of various information elements for machining is difficult and documentation of operations is not carried out with consideration for application characteristics. So, the reincarnation should be accomplished with a minimum of application information. This is one of the reasons why three mandatory inputs were defined in Section 2.
- The rule-based approach (the rules applied in actual practice)
  G-codes contain implicitly the machining experience and know-how of the operator. Hence, a rule-based approach, with rules based on practice by skilled operators, is required. We divide these rules into two types: one is the strongly-applied rule and the other is the weakly-applied rule. The

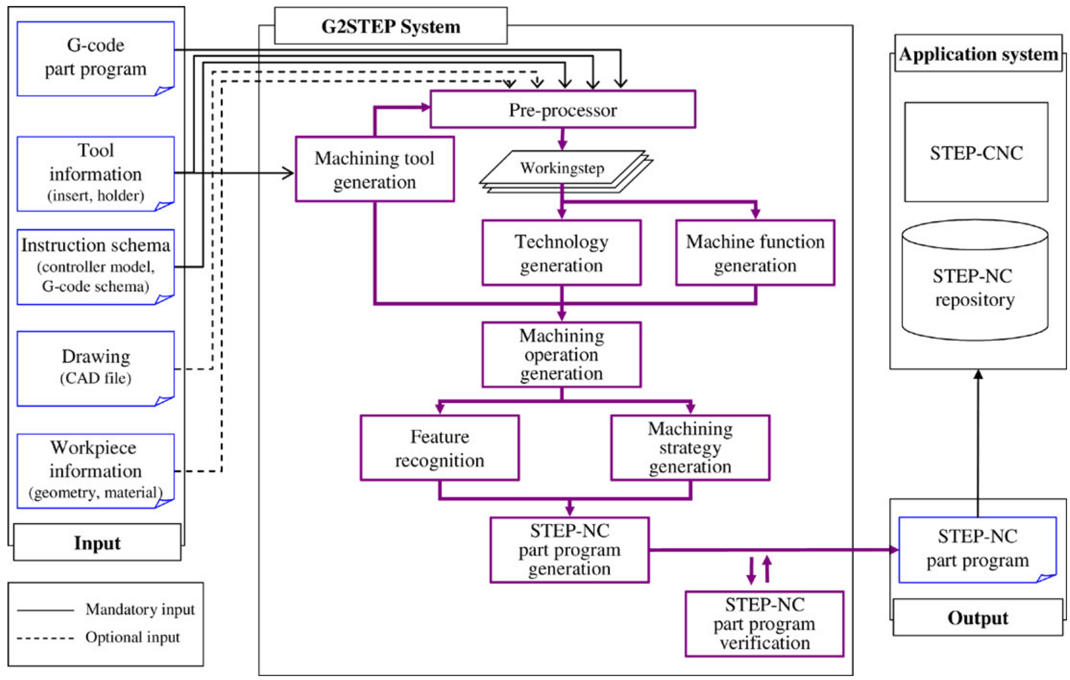


Fig. 2. G2STEP system functional architecture.

former should be applied in all cases for each machining operation and the latter may be applied in most cases, but sometimes it cannot be applied in a few cases. At first, the interpretation is based on strongly-applied rules, and when these prove insufficient, the weakly-applied rules are used. These rules have been established through interviews with skilled operators with 30 years' experience.

- Available and executable STEP-NC information generation

  This system has been developed to have the capability of generating entities and subordinate attributes of features, operations, strategies and others without missing information from machine axis movements. In particular, this system has been designed to generate available and executable ISO 14649 information to the level of Conformance Class 2 — a full set of manufacturing data, especially manufacturing features.
- Accurate reincarnated information

  The final objective is not just correspondence between the two final shapes using two sets of control codes, but also correct STEP-NC workingstep information similar with G-code, because, if this information is inaccurate, the STEP-NC code may have a bad effect on the surface quality of a machined part.

3.2. Functional architecture

Based on the problem clarification in Section 2 and the considerations in Section 3.1, the functional architecture of the G2STEP system is shown in Fig. 2. First, in the pre-processor function, each block of a G-code with other information is interpreted and stored in a data structure in order, and the G-code blocks which have been deemed to carry out the same operation are grouped into the same workingstep. Next, each workingstep has added to its machining tool, machine function, technology, operation, feature and strategy information. Each workingstep object is stored in a workplan and project entity according to executed order, and finally a STEP-NC part program is generated. The STEP-NC part program can be visualized and checked by the viewer. The generated STEP-NC part program is forwarded and utilized by STEP-CNC or stored in a STEP-NC repository. The G2STEP system can be used within a STEP-CNC or as a module of off-line CAM software supporting STEP-NC.

The G2STEP system is made up of nine functions and each function is explained in summary, below:

(1) Pre-processor function: each block of a G-code part program with other information is interpreted and stored in the pre-defined data structure in order. These blocks are divided using some clues as to which G-codes form boundaries to start new machining operations and these blocks are grouped into workingsteps, the basic unit of STEP-NC. See Section 4.1.
(2) Machine tool generation function: the tool holder and insert, expressed by the tool maker's catalogue model, is converted into functional requirements for the model defined in ISO 14649 Part 111 [23] and Part 121 [24].
(3) Technology generation function: technology information such as spindle speed control, feedrate and so on is generated to interpret an S or F-code.
(4) Machine function generation function: machine functions such as coolant on–off, are generated as interpretations of the miscellaneous code (M-code).
(5) Machining operation generation function: the machining operation is mainly generated and divided into roughing and finishing again. This information needs to use information about tool, technology and hence the machining operation generation function follows the functions which generate these. See Section 4.2.

(6) Feature recognition function: the feature profile remaining after a workingstep is generated, and the manufacturing feature, defined by STEP-NC, is recognized by a profile and pre-determined machining operation. See Section 4.3.
(7) Machining strategy generation function: the approach/retract strategy and machining strategy for tool paths are generated. See Section 4.4.
(8) STEP-NC part program generation function: the enriched information of the workingsteps is stored in succession, one workplan entity includes these workingsteps and finally these are included in one project entity, the first interpreted entity in the STEP-NC data model. This information is instanced in the STEP-NC schema and is printed in a physical file format according to the ISO 10303 Part 21 rule [25].
(9) Part program verification function: the STEP-NC part program is verified and modified through visualization. The machined feature profile is shown and each workingstep can be examined and edited through the workingstep editor.

4. Functions for turning machining

The G2STEP system is needed for both turning and milling. However, the rules applied and the algorithms for each function are dependent on the characteristics of the different types of machining. In Section 4, we describe the main functions: pre-processor, machining operation generation, feature recognition and machining strategy generation, together with the rules and algorithms applied for 2-axis (X, Z axis) turning. The information used and generated is based on the ISO 14649 data model, that is, ARM (Application Reference Model) and instruction schema on G-code is based on the FANUC0 series. This is for the clarification of implementation and a similar methodology can be applied for other types of controller models.

4.1. Pre-processor

The Pre-processor function interprets G-code blocks sequentially, stores these in data structures and divides these into groups corresponding to each workingstep using characteristic separator blocks. Note that this function can be applied not only for turning but also for milling because the same pre-processing step is common to both. The information elements stored in the structure are G-code instruction, location coordination, feedrate, spindle speed control, allocated tool number and compensation values, radius in the case of radial interpolation, and so on. In particular, the trajectory vector can be generated through the location change from the pre-movement to post-movement.

Definition 1 (*Trajectory Vector*). A vector for each block, having the direction, length and angle from the start point to the end point for each movement specified by G00, G01, G02 or G03.

Definition 2 (*Characteristic Separator (Block)*). A boundary block dividing G-code into workingsteps. The indicators of a separator block are tool change, spindle or feedrate change, the change of machining direction or machining area.

Definition 3 (*Interpolation Type Block or Interpolation Type Vector*). A block or a vector given by the machining instruction type G01, G02 or G03. G01 is for line interpolation, G02 and G03 are for radial interpolation.

Definition 4 (*Stroke*). A cycle composed of approach, machining and retract tool paths for removal of one layer of a workpiece. For example, the roughing operation can be made up of several strokes to remove all material to a specified allowance.

Workingsteps, sectioned by the characteristic separators, have two types. One is the setup workingstep and the other is the operation workingstep. A setup workingstep means the workingstep which is made up of only coordination setup, absolute relative instruction, feedrate per minute or revolutions, spindle speed control, coolant on–off. We can get only information about technology and machine function by checking S, F, M codes in this workingstep. This information has an effect on the subsequent operation workingsteps, and defines a sort of operational environment. In contrast, the operation workingstep is a real machining workingstep made up of blocks having tool movement instructions. An operation workingstep can be mapped into one machining operation. The blocks included in this are rapid travels like G00 or interpolation type blocks. We thus need to trace their movements. In turning or milling machining, cycle codes, the automatic tool path generation and execution of one machining operation in a controller can be present to facilitate effective editing of a part program. These cycle codes are commonly used in practice, but it is easy to generate the information of an operation, manufacturing feature and strategy and so on, so the whole machining information about cycle codes can be generated by a separate method only for those.

Fig. 3 shows the procedure of the Pre-processor function. Initially, this function reads a part program and interprets it according to the G-code schema of a controller. The blocks with only technological information are divided into the setup workingsteps. The first division is done using characteristic separators with a T-code, that is, when tool change occurs. The reason why G-codes are firstly grouped into units with the same T-code is that, in general, an operator makes a part program considering the minimum number of tool changes. Machining methods are also dependent on the type of tool, so the interpreting methods vary according to the allocated tool. The characteristic separators which change cutting speed or feedrate divide the first division block groups into the second division groups. Because a change of cutting speed or feedrate means a change in the quality of the machined surface, a skilled operator seldom changes these during one operation. So, a change in cutting speed or feedrate implies another new operation is beginning. Next, the characteristic separators which change the machining direction provide an indication of a third type of division into workingsteps. Even though the same cutting speed, feedrate and tool are used, another machining operation can occur if the machining direction

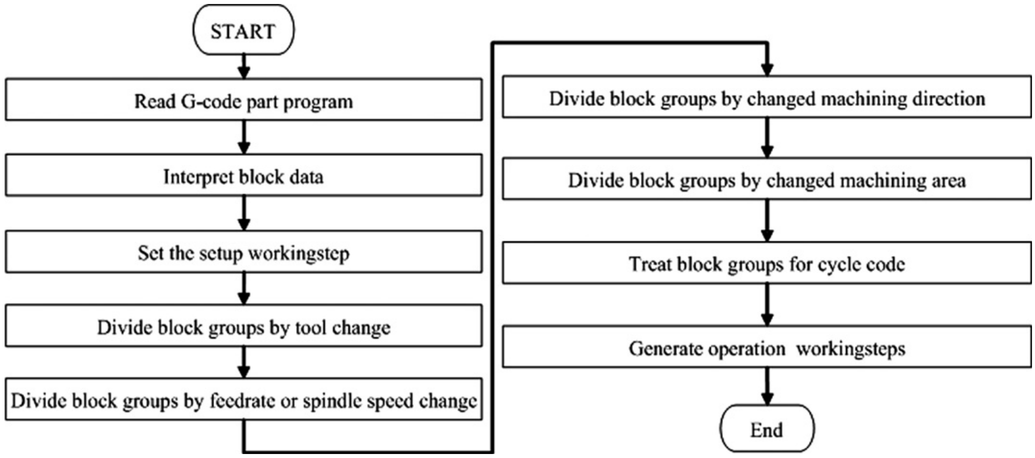


Fig. 3. Procedure of pre-processor.

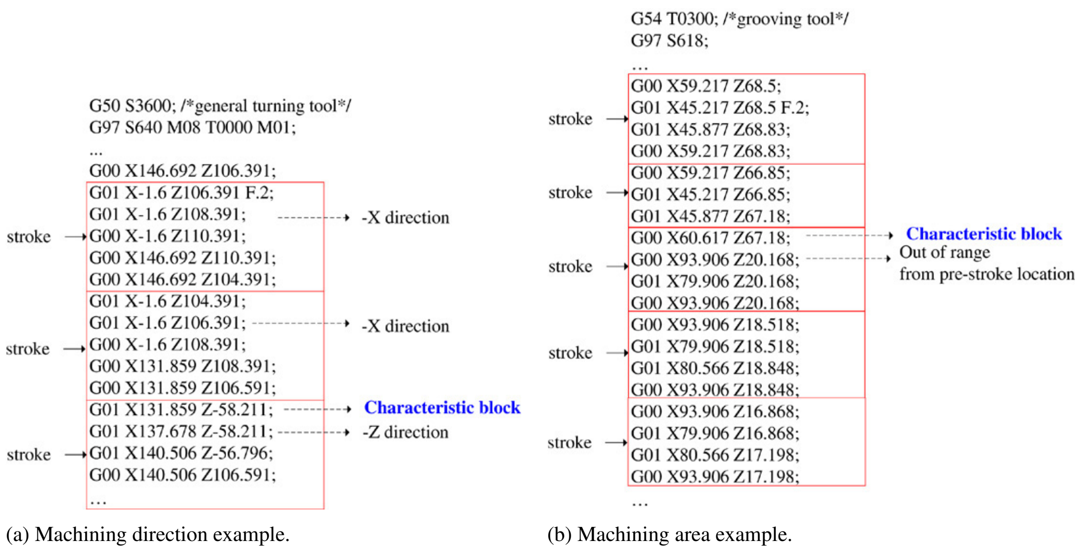


Fig. 4. An example of characteristic separator determination for machining direction and area.

changes, for example, if there is a facing operation machining perpendicular to the Z axis and then a contouring operation parallel to the Z axis. Also, the characteristic separators which indicate positions in a new machining area divide the G-code blocks into the final operation workingstep, even though the other three conditions are the same.

The two determination methods of characteristic separators for changing machining direction and machining area are as follows.

- Method for the determination of the characteristic separator by change of machining direction (see Fig. 4(a)).

  If the direction of the first interpolation type vector of stroke n is similar to the direction of that of stroke $n-1$, $(n = 2, 3, \ldots)$.

  Then, stroke n is in the same direction as stroke $n-1$.

  Else if stroke n is in a different direction from stroke $n-1$ and the first rapid travel block of stroke n is set to the characteristic separator.
- Method for the determination of the characteristic separator by change of machining area (see Fig. 4(b)).

  If the start and end locations of the first interpolation type block of stroke n are within range of the cutting edge length of the allocated tool at the location of the $n-1$ stroke,

  Then, stroke n is in the same area as stroke $n-1$.

  Else if stroke n is in a different area from stroke $n-1$ and the first rapid travel block of stroke n is set up to the characteristic separator.

The reason why these methods can be applied is that the type of machining operation determines the direction of the first interpolation type vector of strokes for shortest-time machining by both an operator and CAM software. Also, the tool paths have a uniform direction pattern, when removing several layers of the workpiece in one operation. If these procedures end the work of a division then all blocks of a part program are finally grouped into an operation workingstep unit.

Fig. 5 gives an example of how part program blocks are grouped into setup or operation workingsteps, following the procedure of Fig. 3. Firstly, blocks having only setup information are divided into setup workingsteps (lines N10–N11, N41–N42). The remaining blocks are separated

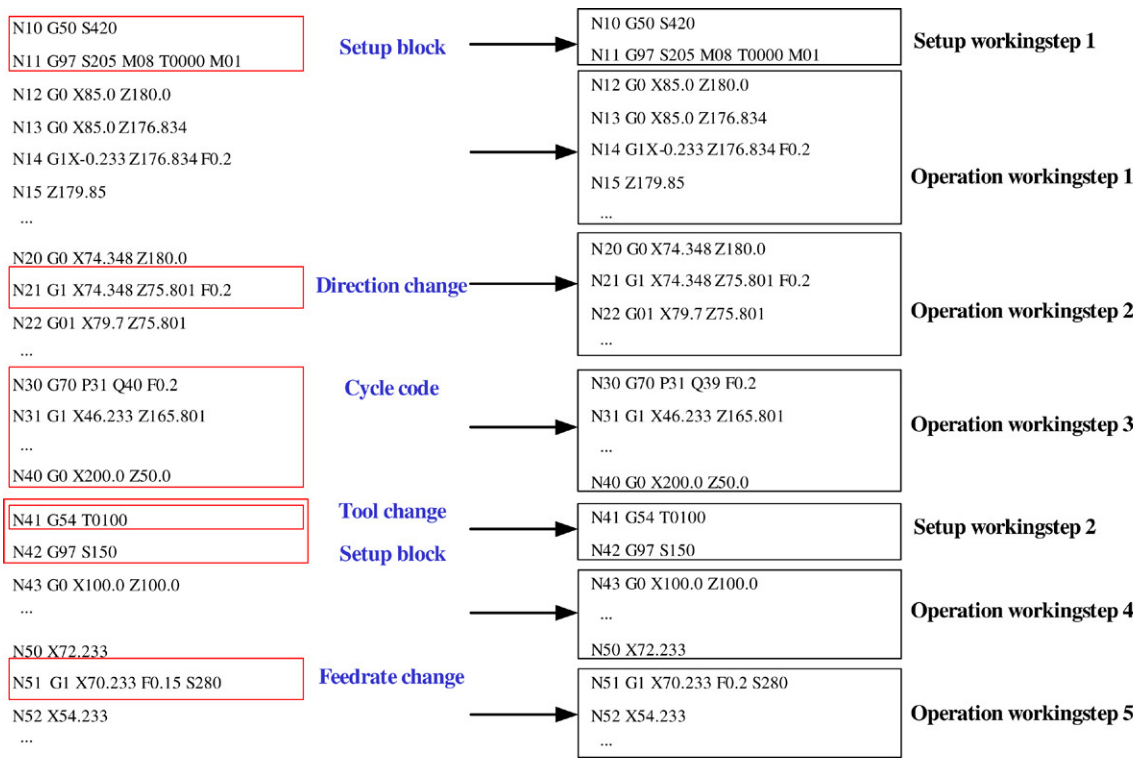


Fig. 5. An example of workingsteps composed of block groups.

by the characteristic separator changing T-code (line N41). The characteristic separator (line N51) changing the feedrate or spindle speed splits the separated blocks further. The characteristic separator (line N21) changing the machining direction divides block groups into the final workingsteps. If a cycle code exists, a cycle code (lines N30–N40) is treated. These subdivided workingsteps are defined as operation workingsteps.

4.2. *Machining operation generation*

This function generates the type of machining operation and its attributes for an operation workingstep. This function is first applied to the operation workingsteps defined in the Pre-processor. This is because the machining operation information is relevant when deciding the type of feature and strategy. The criteria for the determination of the type of operation are mainly the type of the allocated tool, the directions of the interpolation type blocks, feedrate and spindle speed. Examples of the rules applied are shown in Table 3. The rules include the strongly-applied rules, in bold letters, and the weakly-applied rules, in normal letters. The types of machining operation that can be generated are: *facing*, *external/internal contouring* (actually these two operations are not separated in STEP-NC), *grooving* (in the turning model) and the *drilling_type_operation* (in the milling model). We exclude *threading* (formalized entity in ISO 14649) and *knurling* operations in the turning model because these two operations are special machining.

Fig. 6 represents how we can define the type of machining operation for each operation workingstep. Initially we get information about the operation workingstep. We find out available operation types by the allocated tool. Later, the exact operation type is determined by machining direction or machining area. The machining operation is classified into roughing and finishing in detail. The criteria for this classification are the use of a finishing tool, the comparison with the technology of the roughing operation, the existence of a roughing operation in the same machining area. Experienced operators have a tendency to increase spindle speed and lower feedrate for finishing compared with roughing, because they want to smooth and elaborate surfaces of a part through the finishing operation. Following these procedures, not only the type of machining operation but also some subordinate attributes can be generated.

4.3. *Feature recognition*

We generate the feature geometry and map this into STEP-NC manufacturing features for a workingstep according to the type of machining operation and the type and geometry of the tool. It is important to note that a feature means not the design feature from CAD modelling but the machining feature, recognized from the workpiece remaining after the operation. The procedure of feature recognition is shown in Fig. 7. Initially, we set the remaining workpiece area for the present operation workingstep, then decide which cutting edge of a tool is in contact with the workpiece and the machining area is calculated from the interpolation type block. We subtract all of the machining areas from the workpiece by Boolean operations. Finally, we extract the feature profile and recognize to which manufacturing feature it can be mapped.

(1) Set workpiece area

We set the remaining workpiece area before the present workingstep is executed. The method to define the workpiece

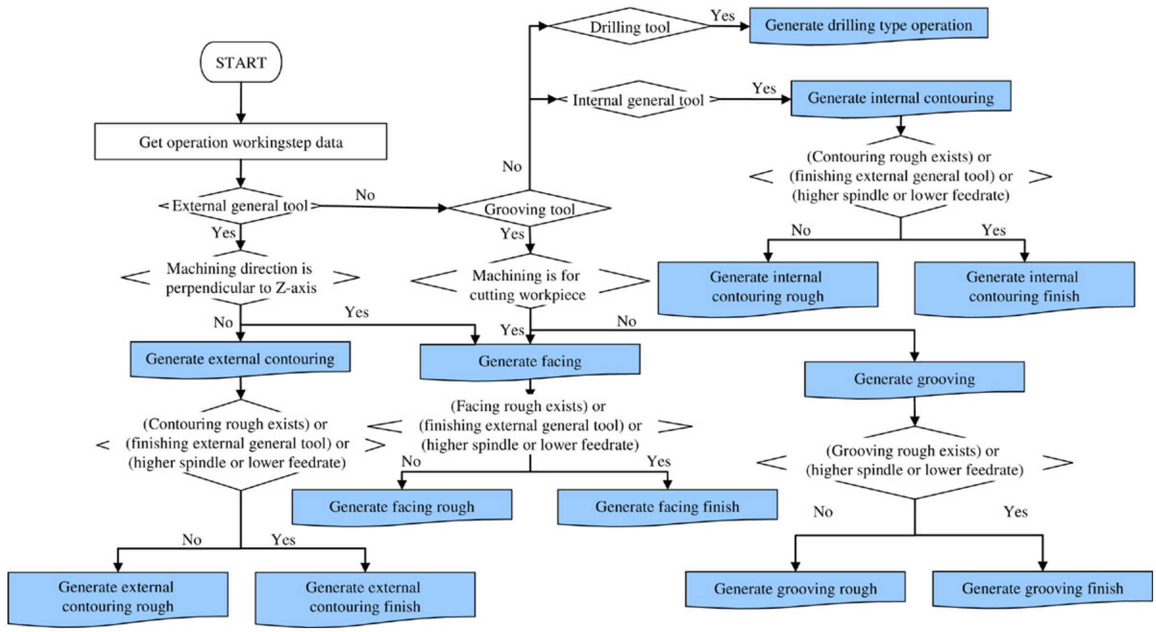


Fig. 6. Procedure of machining operation generation.

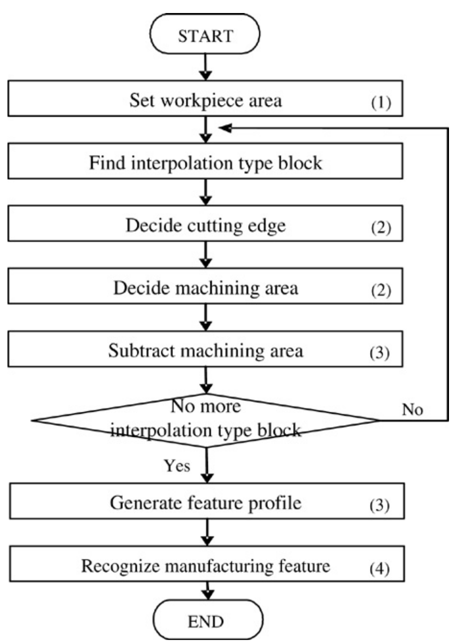


Fig. 7. Procedure of feature recognition.

area for the first operation workingstep differs from that used for the other operation workingsteps. In the case of the first operation workingstep, if the optional workpiece information has been input by an operator, the initial workpiece area is set using this information. If not, then we need to make some assumptions. When the first operation is *facing*, a vertex – the Z value is the plus *cutting_edge_length**cos(*end_cutting_edge_angle*) with the Z value of the pre-movement point of the very first interpolation block, the X value is the same as the X value of the pre-movement of that – is generated with position in the ZX plane. When the first operation is *contouring*, then a vertex – the X value is given by subtracting *cutting_edge_length**sin(*side_cutting_edge_angle*) from the X value of the pre-movement point of the very first interpolation block, the Z value is the same as the Z value of the pre-movement of that – is generated. The Z value of another vertex can be generated by searching for the minimum value in the whole part program and the X value set to the above X value. We can make one line connecting two vertices and sweep this line along the Z axis to generate a rectangular area, which is implicitly a cylindrical part that is assumed to be the initial workpiece. In the case of the remaining operation workingsteps, the volume remaining after the execution of each operation workingstep is used as the new workpiece area for the subsequent operation, as shown in Fig. 8.

(2) Decide cutting edge & machining area

We decide the cutting edge considering the direction of the trajectory vector, the type and geometry of the tool. Using tool geometry, we can determine the cutting edge – the insert cutting surface including this cutting edge and its contact with the workpiece – when the interpolation movement is applied. Fig. 9 shows the cutting edge decided by the tool type and the direction of the trajectory vector. For internal/external turning tools, the cutting edge can be the side cutting edge consisting of *side_cutting_edge_angle* (Fig. 10(b)), the end cutting edge consisting of *end_cutting_edge_angle* (Fig. 10(d)) and both cutting edges (Fig. 10(c)), independently from *hand_of_tool*. For a grooving tool or a drilling tool, the cutting edge is the tool tip.

After the determination of the cutting edge, we can get a machining area such as that shown in Fig. 11(a), as the cutting edge traversal area from the start point to the end point of an interpolation type vector. Even though, for the general turning tool with both edges or the drilling tool, there are two cutting edges, the same method is applied for each cutting edge (Fig. 11(b)).

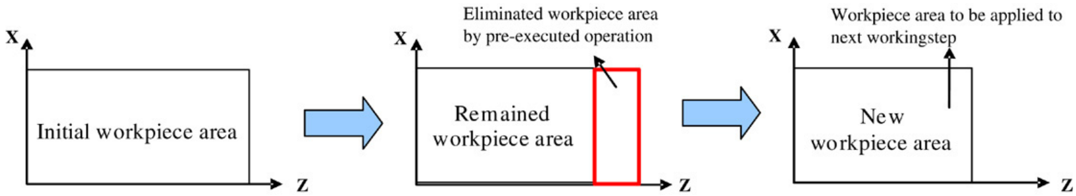


Fig. 8. Setting a new workpiece area for a subsequent operation.

Cutting edge	Tool type	Trajectory vector direction range	Example figure
Side cutting edge	General turning tool	180° - side cutting edge angle ~ 180° + end cutting edge angle	Figure 10(b)
End cutting edge	General turning tool	360° - side cutting edge angle ~ end cutting edge angle	Figure 10(d)
Side cutting edge & End cutting edge	General turning tool	180° + end cutting edge angle ~ 360° - side cutting edge angle	Figure 10(c)
Tool tip edge	Grooving tool		
Tool tip edge	Drilling tool		

Fig. 9. Decision about cutting edge by tool type and trajectory vector direction.

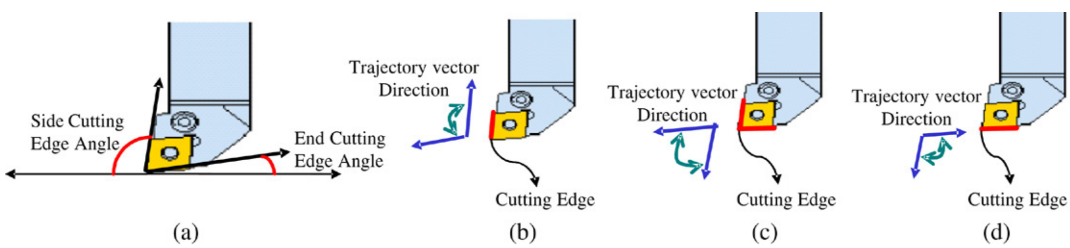


Fig. 10. Cutting edge of general turning tool by cutting direction. (a) General turning tool and cutting edge angle, (b) when side cutting edge, (c) when both edges are cutting edges, (d) when end cutting edge.

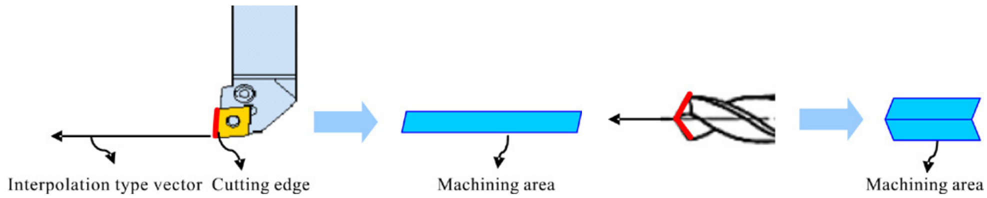
Table 3
Rules applied in each machining operation

Machining operation	Rules	Machining operation	Rules
Facing rough	**The use of general turning tool** **The machining movement is perpendicular to Z axis** The use of roughing turning tool or roughing facing tool The use of grooving tool for cutting of the material The first machining operation Z values of interpolation type blocks are decreased	Facing finish	**The use of general turning tool** **The machining movement is perpendicular to Z axis** The use of finishing turning tool or finishing facing tool Faster spindle speed or lower feedrate than roughing
External contouring rough	**The use of general turning tool** The use of roughing turning tool X values of interpolation type blocks are decreased	External contouring finish	**The use of general turning tool** **The contour turning strategy** The use of finishing turning tool Faster spindle speed or lower feedrate than roughing
Grooving rough	**The use of grooving tool** The machining direction is perpendicular to Z axis or X axis	Grooving finish	**The use of grooving tool** Faster spindle speed or lower feedrate than roughing
Internal contouring rough	**The use of internal turning tool** The post-operation of drilling operation The use of internal roughing turning tool X values of interpolation type blocks are increased	Internal contouring finish	**The use of internal turning tool** **The contour turning strategy** The use of internal finishing turning tool Faster spindle speed or lower feedrate than roughing
Drilling type operation	**The use of drilling tool** The center drilling operation for the pre-operation The multistep drilling for easy removal of chips		

⟨Bold: strongly-applied rule, normal: weakly-applied rule⟩

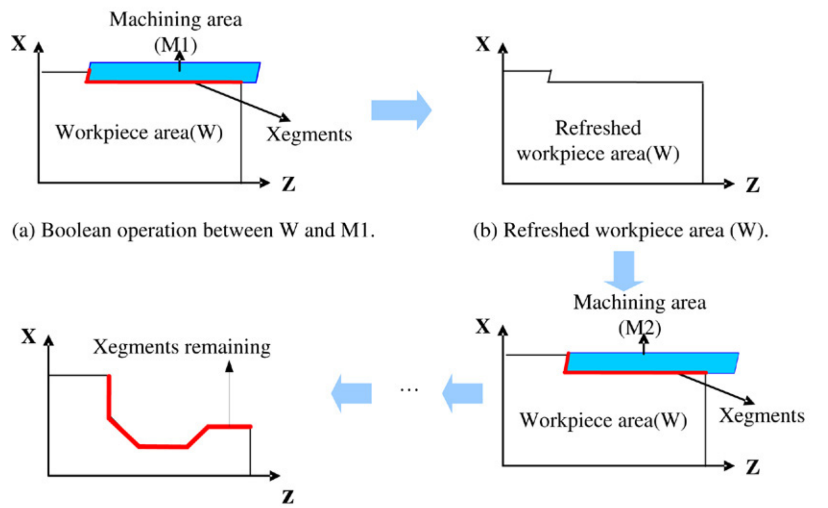
(3) Subtract machining area & generate feature profile

The feature profile of the remaining workpiece is generated using a Boolean subtract operation [26] of all machining areas generated from tool movements in an operation workingstep and the input workpiece area. As shown in Fig. 12, the Xegments [27], which are the intersection curve segments



(a) Machining area with one cutting edge. (b) Machining area with two cutting edges.

Fig. 11. Cutting edge passed machining area by interpolation type vector.



(a) Boolean operation between W and M1. (b) Refreshed workpiece area (W).
(d) Generated feature profile. (c) Boolean operation between W and M2.

Fig. 12. Generation of feature profile. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Manufacturing feature mapping matrix

Machining operation	Number of profile segment		Type of profile segment		Inclination of profile segment		Mapping manufacturing feature
	1	More than 2	Line	Arc	Not exist	Exist	
Facing	O		O		O		Circular face
Facing	O		O			O	Revolved flat
Facing		O					General revolution
Contouring	O		O		O		Outer diameter
Contouring	O		O			O	Revolved flat
Contouring	O			O			Revolved round
Contouring		O					General revolution
Grooving		O	O				Groove
Drilling		O	O				Round hole

between two areas, are extracted and retained. After subtracting the machining areas of all interpolation type blocks from the workpiece area, the Xegments [27] remaining give the feature profile.

(4) Recognize manufacturing feature

Workpiece shapes are mapped onto the set of manufacturing features defined in the STEP-NC data model. The feature profile includes the types and locations of profile segments. STEP-NC manufacturing features can be classified by the segment characteristics. Also, even the same feature profile can be defined as different features for convenience. For instance, a plane surface perpendicular to the Z axis for a *facing* operation can be represented variously as *outer_diameter*, *revolved_flat* and *circular_face*. However, for uniqueness, we classify the extractable feature and map it onto the manufacturing feature according to the feature profile and the machining operation type. Table 4 shows the manufacturing features mapped by operation and the feature profile characteristics. For example, if the machining operation is *contouring* and the number of segments is five like Xegments [27] remaining – red bold straight lines – of Fig. 12(d), then the manufacturing feature is mapped onto *general_revolution* using the feature mapping matrix.

4.4. Machining strategy generation

The machining strategy generation function generates approach/retract and machining strategies. The main criteria for the determination of these strategies are the type of machining operation, the type of tool, interpolation type blocks, rapid

Table 5
Rules applied in each machining strategy

Machining strategy	Rules
Unidirectional turning	**Back path moved by rapid travel having the reverse direction of the feed direction** Lift action existence for the next feed movement Stepover path moved perpendicular to the feed direction by rapid travel to the next machining start point
Bidirectional turning	**No back path direction** **Continuous interpolation movements** **The use of neutral hand type tool**
Contour turning	**Tool paths following the feature profile** **The numbers of interpolation movements more than 2 continuously**
Grooving strategy	**The use of grooving tool when grooving operation** Dwell time existence The existence of the rapid travels having the reverse direction of the interpolation movements

⟨bold: Strongly-applied rule, normal: Weakly-applied rule⟩

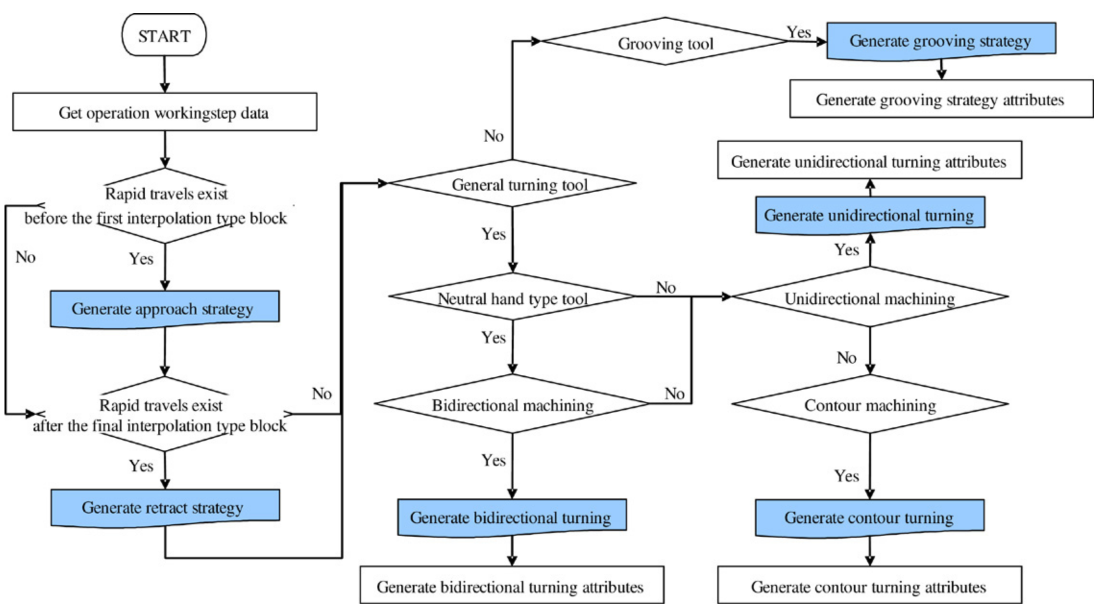


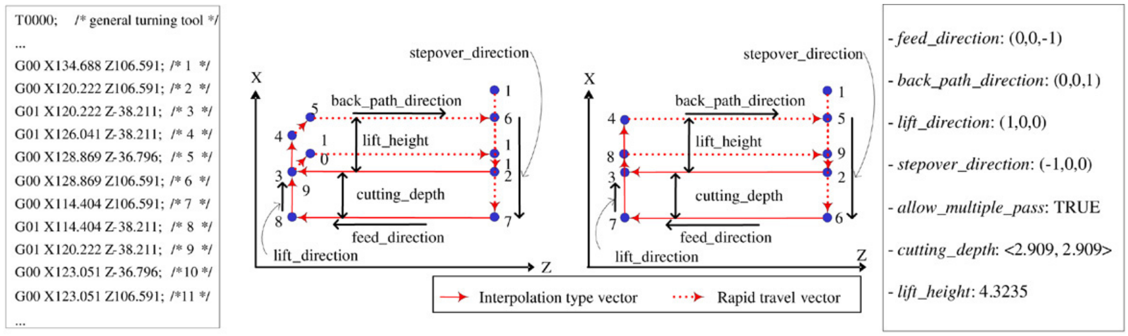
Fig. 13. Procedure for machining strategy generation.

travel blocks and the combination relationship of both of these. Table 5 shows the rules applied for the machining strategy determination in the same manner as for the machining operation.

In Fig. 13, the procedure for determining the strategies is represented. If rapid travel blocks exist before the first interpolation type block in an operation workingstep, then those can be defined as the approach strategy. If rapid travels exist after the last interpolation type block, the movement of those define the retract strategy in the same way as for the approach. Depending on which rules of Table 5 are applied to tool paths, *bidirectional_turning*, *unidirectional_turning*, *contour_turning* and *grooving_strategy* can be determined. For example, if using the external turning tool having a neutral hand type and interpolation type blocks have one direction and the reverse direction continuously, then *bidirectional_turning* and its attributes are generated. The *drilling_type_strategy* is out of scope, because the information can be optional and mainly related not to tool movements but to cutting speed or feedrate.

It should be noted that subordinate attributes are dependent on the type of machining strategy, so methods suitable for each strategy are required. For instance, in the part program of Fig. 14(a), the *unidirectional_turning* operation can be set because the general turning tool is used, the interpolation type blocks have Z-axis direction and rapid travels in the reverse direction exist. As shown in Fig. 14(d), attributes of machining strategy can be generated.

It is worth mentioning that not all blocks of a part program have an ordered set of patterns, as those in Fig. 14 do. In particular, a part program produced by CAM software can have very complicated patterns, so it is not easy to generate homogeneous values for attributes. Also, STEP-NC requires only one value for each attribute except for some list type attributes. Because of this we select and decide the representative values for attribute instances from a special stroke in an operation workingstep. In a few cases, STEP-NC tool paths don't coincide with G-code tool paths due to the STEP-NC data structure as in Fig. 14(b) and (c). However,



(a) G-code example. (b) Tool path on G-code. (c) Tool path on STEP-NC. (d) STEP-NC strategy example.

Fig. 14. An example of machining strategy attributes generation.

discordant tool paths are all related to rapid travels, so this does not have an effect on machining quality.

5. Examples

5.1. Development of a prototype G2STEP system

Based on the considerations and functions described previously, a prototype system for 2-axis turning machining called G2STEP was developed. This system was written in C++ and operates on a Windows platform. Parasolid was used for the geometric modelling kernel and the graphical user interface was coded mainly using OpenGL. As shown in Fig. 15, the following example presents the complex turning example in the appendix of ISO 14649 Part 12 [28]. The workpiece has 9 machining operations: facing rough, external contouring rough, external contouring finish, grooving rough, grooving finish, center drilling, multistep drilling, internal contouring rough and finish. First, the tool numbers and information attached to the turret are set up (Fig. 15(a)). In this example, tools made by TaeguTec, Inc. are used. Compensation values for each tool are input (Fig. 15(b)). If a CAD file exists, it is opened (Fig. 15(c)). This CAD information is not extracted and is visualized only for the user's convenience. If the workpiece information exists, the shape and material are defined (Fig. 15(d)). The type and the controlled axis of the controller are set (Fig. 15(e)). The FANUC0 series for a 2-axis turning machining is used. The G-code part program for right setup is loaded (Fig. 15(f)). This part program was generated by commercial software, EdgeCAM (V8.0). When these operations are finished, as shown in Fig. 15(g), this part program generates 9 operation workingsteps, instanced as C++ classes in the system. Additionally, Fig. 15(g) shows STEP-NC information for verification and visualization. Finally, an ISO 14649 physical file is generated, as shown in Fig. 15(h).

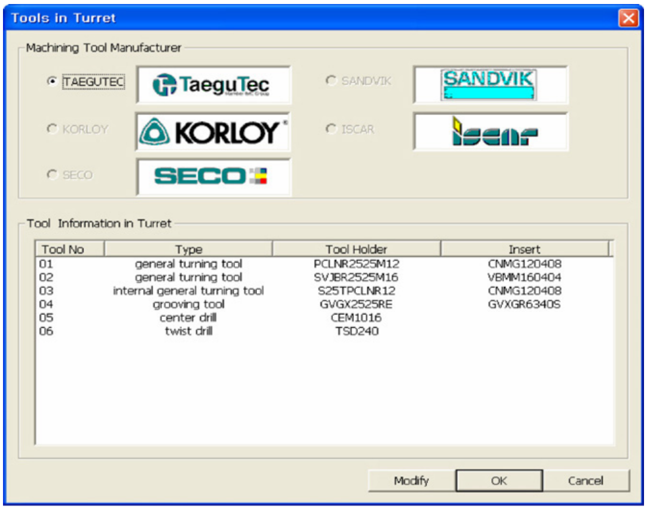
5.2. Verification

We compare the final part machined using the output STEP-NC part program with the final part machined using the original G-code part program on the Virtual NC software from Delmia, Inc. However, commercial controllers or applications having the ability to interpret a STEP-NC part program and to execute machining directly have not yet been launched. In addition, the Virtual NC requires post-processed G-code part programs, so we use the CES module of TurnSTEP [16]. TurnSTEP, developed by our research team, is the STEP-NC based autonomous and intelligent CAPP/CAM system and CES has a function to convert a STEP-NC part program into G-code for Virtual NC or conventional NC.

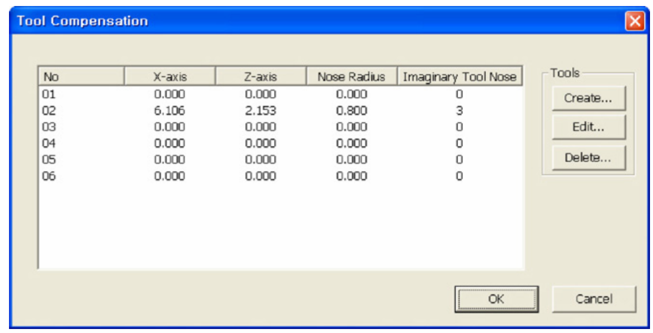
Fig. 16 presents the analysis results of the difference between the two final parts using a verification scenario. In Virtual NC, when the shape difference between two parts is larger than a defined tolerance, then over cutting or under cutting is shown using a colour representation according to the extent of the tolerance. The result of the example in Section 5.1 shows that the two parts are identical to within the tolerance limits of 0.01 mm.

6. Conclusions

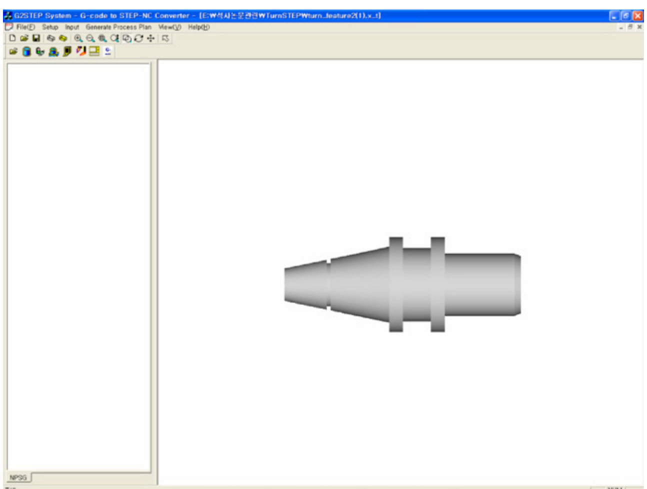
In this paper, we presented the system functional architecture and algorithms for converting G-code into STEP-NC. We have also verified the accuracy of the algorithms by developing a prototype G2STEP system. The information including planning level was reincarnated from the minimum requirement condition. For this, a rule-based approach is used, based on experience from skilled operators. Such methods are needed when STEP-NC is employed in real industrial practice, that is, when the second level of the STEP-Manufacturing Roadmap has been achieved. The G2STEP system can contribute to time and cost reduction by automating conversion of G-code into STEP-NC. This is needed to overcome the problem of legacy code preserved by companies when STEP-NC controllers become widely available. Because a STEP-NC part program can be generated easily by operators skilled in using G-code, the early adoption of the STEP-Manufacturing environment can be realized. In addition, accumulation of know-how from actual machining can be included. As a result, this system is an indispensable tool for the implementation of a STEP-NC application and repository. Future work on these algorithms includes developments to enhance robustness through tests and for more controllers and more examples of real industrial parts. This is needed because different and more complicated methods are widespread and these vary according to the operator's personal preferences. Also, similar research



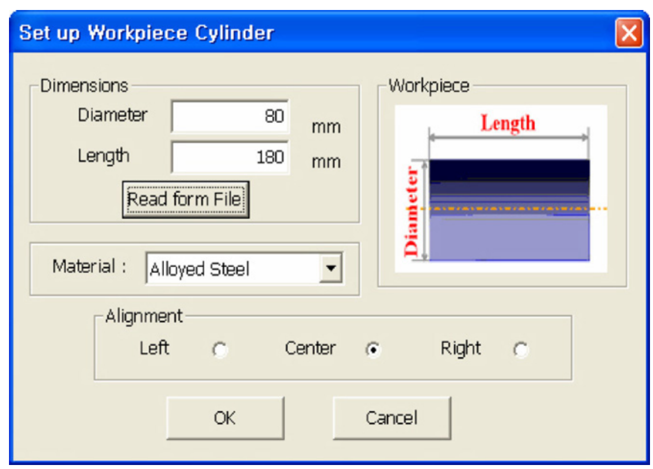
(a) Tools information in turret.



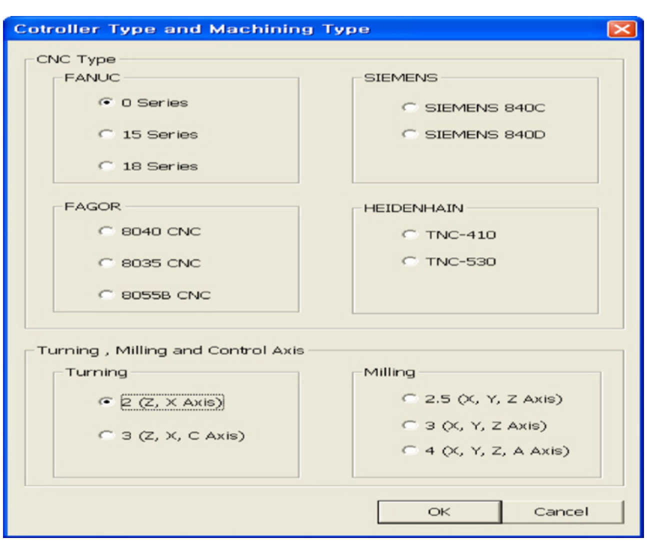
(b) Tool compensation.



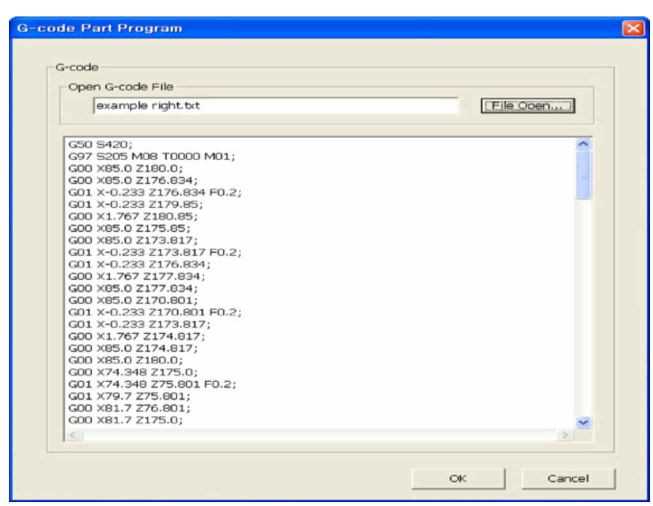
(c) CAD file.



(d) Workpiece information.

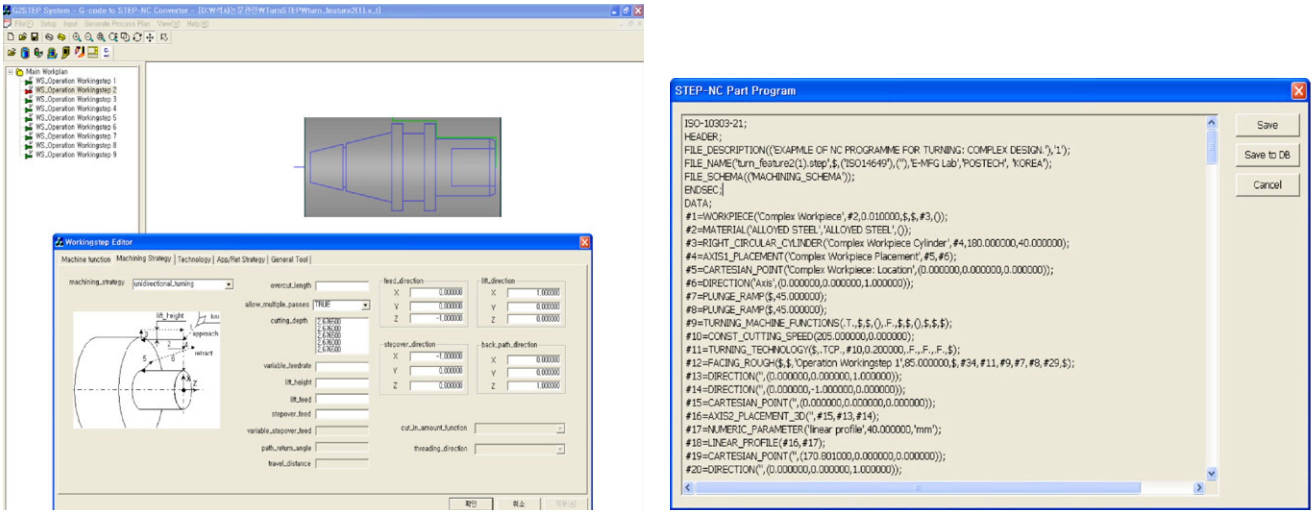


(e) Controller type & control axis.



(f) G-code part program.

Fig. 15. Screen examples of G2STEP system.



(g) STEP-NC information visualization.

(h) STEP-NC part program.

Fig. 15. (*continued*)

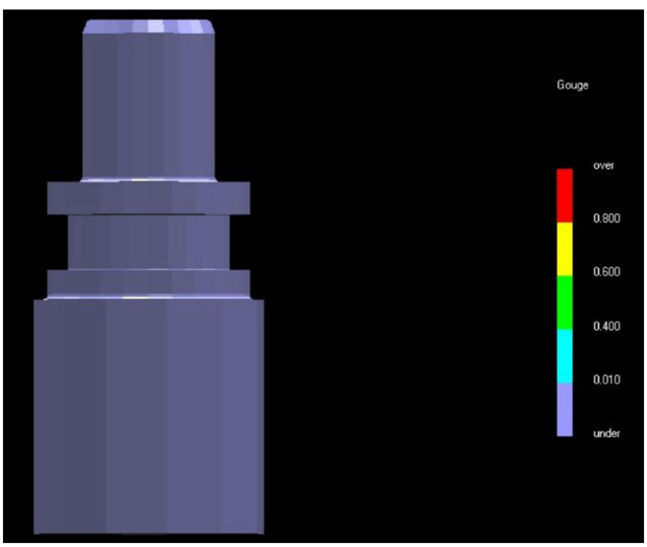


Fig. 16. Machined part geometry analysis.

into milling applications, multi-axis and complex machining has to be developed and applied. The G2STEP system is an important step along the road to handling the legacy problem of NC applications.

Acknowledgements

This research was supported by a grant from the National Research Laboratory for STEP-NC Technology by MOST (Ministry of Science and Technology) in Korea. The authors acknowledge Yeung-Pyo Park, who has 30 years' experience in industry and POSTECH and his colleagues, at Central Machine and Facilities Shop Technical Support Department in POSTECH, for their support in providing advice and rules.

References

[1] Freeman B. The STEP manufacturing suite. In: White paper presented on ISO TC184/SC4 meeting. 2001.
[2] Albert M. STEP-NC — the end of G-codes. Mod Mach Shop 2000;1: 70–80. Online.
[3] Hardwick M. Justifying the STEP-NC savings. In: Proceedings of fourth MDICM IRB meeting. 2001.
[4] Leyrich M. The ultimate step. Am Mach 2001.
[5] Albert M. Feature recognition — the missing link to automated CAM. Mod Mach Shop 2001;15. Online.
[6] Maniscalo M. *E*-capable CNC gains ground. Inject Mod Mag 2001.
[7] Teresko J. The next STEP. Ind Week 2001.
[8] Hardwick M, Loffredo D. STEP into NC. Manuf Eng 2001.
[9] Suh S, Lee B. STEP-Manufacturing roadmap. In: Proceedings of Korea CAD/CAM conference. 2004.
[10] Suh S, Cho J, Hong H. On the architecture of intelligent STEP-compliant CNC. Int J Comput Integr Manuf 2002;15(2):168–77.
[11] International Standards Organization. ISO14649. Part1: Overview and fundamental principles. 2003.
[12] Glantschnig F. STEP-NC is reality. In: White paper presented on ISO TC184/SC4 meeting. 2000.
[13] Hardwick M. US STEP-NC implementation. In: White paper presented on ISO TC184/SC4 meeting. 2001.
[14] Suh S. Korea STEP-NC. In: White paper presented on ISO TC184/SC4 meeting. 2001.
[15] Darrol H. JPL and STEP-NC: A work in process. In: White paper presented on 6th IRB meeting. 2003.
[16] Suh S, Chung D, Lee B, Shin S, Choi I, Kim K. STEP-compliant CNC system for turning: Data model, architecture and implementation. Comput Aided Des 2006;38:677–88.
[17] Vankhai N, Jacques R, Willy M. STEP-NC: A new intelligent interface from CAD to numerical controller. In: White paper presented on UTGV workshop. 2004.
[18] Weck M, Wolf J, Kiritsis D. STEP-NC — The STEP compliant NC programming interface evaluation and improvement of the modern interface. In: White paper presented on proceedings of the IMS project forum. 2001.
[19] Xu X. Realization of STEP-NC enabled machining. Robot Comput-Integr Manuf 2006;22:144–53.
[20] Newman S, Allen R, Rosso R. CAD/CAM solutions for STEP-compliant CNC manufacture. Int J Comput Integr Manuf 2003;16(7–8):590–7.
[21] STEPTools Inc. ST-Developer R11 beta applications. Operation manual. 2005.
[22] International Standards Organization, ISO 10303. AP238: Application protocol — application interpreted model for computer numeric controls. 2004.
[23] International Standards Organization, ISO 14649. Part111: Tools for milling machine. 2004.

[24] International Standards Organization, ISO 14649. Part121: Tools for turning machine. 2005.
[25] International Standards Organization, ISO 10303 Part21: Implementation methods — clear text encoding of the exchange structure. 2002.
[26] Aristides R, Herbert V. Boolean operations in solid modeling: Boundary evaluation and merging algorithms. Proc IEEE 1985;73(1):30–44.
[27] Lee K. Principles of CAD/CAM/CAE systems. 1st ed. Addison Wesley; 1999.
[28] International Standards Organization, ISO 14649. Part12: Process data for turning. 2005.