Int J Adv Manuf Technol (2002) 19:727–735
Ownership and Copyright

The International Journal of Advanced Manufacturing Technology

A Framework for an Intelligent CNC and Data Model

S.-H. Suh and S.-U. Cheon

National Research Laboratory for STEP-NC Technology (NRL-SNT), School of Mechanical and Industrial Engineering, POSTECH, San 31, Hyoja-dong, Pohang, Korea

In this paper, we present a conceptual framework for designing and implementing an intelligent CNC system. The architecture and functional modules are derived from requirement analysis, and they are designed to have a thinking capability before, during, and after the execution of the manufacturing task, so that the assigned task can be effectively executed while dealing with unexpected changes occurring on the shop floor. Also, the data model for supporting the architecture is addressed based on the STEP-NC data model or ISO 14649. Specifications of databases and the operational scenario together with implementation issues are provided. The framework presented can be used as a paradigm for STEP-compliant CNC.

Keywords: Autonomous machining control; Holonic manufacturing system; Intelligent CNC; ISO 14649; STEP-compliant CNC; STEP-NC

1. Introduction

As the brain for industrial machinery, computer numerical control (CNC) is the core element in a modern manufacturing system. CNC technology is highly complex, requiring technologies from various domains. In spite of great technological achievement, contemporary CNC still requires further improvement to overcome its drawbacks:

1. It is an executing mechanism without intelligence.
2. It uses a low-level language, the so called M & G codes (ISO 6983) as an input.
3. Its architecture is vendor specific and black-box styled proprietary without allowing user access.

Therefore, the next generation CNC is required:

1. To use a high-level programming language for seamless integration of the CAD-CAM-CNC chain.
2. To be multi-functional, intelligent and autonomous.
3. To have an open architecture with software-based implementation technology.

These requirements should be included in the next generation CNC.

A new and comprehensive data model for the programming language formalised as ISO 14649 is being developed by ISO TC 184 SC1 WG7 [1]. ISO 14649 describes an interface between CAM and CNC to support the direct use of computer-generated product data based on STEP (ISO 10303) for creating exchangeable and workpiece-oriented data models for CNC controllers. It is currently available in DIS-ballot version, with its final version to be completed in the near future. Upon completion, it will be a new CNC language replacing ISO 6983. STEP-NC is an acronym of ISO 14649 and is extended to form a new CNC that can carry out various intelligent functions based on ISO 14649 [2]. For intelligent control, however, the new data model must be updated [3].

Several schemes have been proposed for NC architecture. The holonic numerical controller (HNC) [4,5] is based on the paradigm of a holonic manufacturing system (HMS) (HMSFPD 1995 [6–8] where NC holons are characterised by autonomy and cooperation. TRUE-CNC [9] is a comprehensive CNC system with CAD, CAPP, CAM, CNC, monitoring, and inspection functions. Brouer and Weck [10] described the basic concepts of an autonomous production cell with a special focus on its programming interface.

The industrial aspect of CNC architecture is related to implementation methodology. Most CNCs developed up to the present time are of "closed" architecture, which do not allow users to access the internal functions for any modification. Developing an "open" CNC architecture has been one of the main concerns for both CNC makers and users since the mid 1980s, but it was not until the early 1990s that a few commercial products were investigated by some leaders in the CNC industry. Liu [11] emphasises the requirement for open architecture to realise agile production and autonomous CNC control. However, a practically efficient open-CNC is still under development with different architectural schemes, such as OSACA in Europe, OMAC in the USA, and OSEC in Japan [12].

Correspondence and offprint requests to: Professor S.-H. Suh, National Research Laboratory for STEP-NC Technology, School of Mechanical and Industrial Engineering, POSTECH, San 31, Hyoja-dong, Pohang 790-784, Korea. E-mail: shs@postech.ac.kr. http://stepnc.postech.ac.kr

This paper aims at deriving a conceptual framework for an intelligent CNC in terms of:

1. The requirement analysis from the perspective of the shop floor,
2. Functional architecture to meet the requirements,
3. Data models supporting the architecture,
4. The operational scenario,
5. Implementation issues.

2. Requirement Analysis

In designing the details of an intelligent CNC, various aspects should be taken into consideration. The shop floor perspective is a guideline for defining the control functions (function-level). Also important are the data interface and information contents in order to support the control functions (data-interface-level), and finally, the implementational aspect (implementation-level). The requirements for each level are summarised in the following.

2.1 Function-Level Requirements

Function-level requirements are related to the activities of CNC to carry out successfully the mission given to it. The principal function is to control the machine tool so that the desired shape is produced accurately by the machining operation. To be intelligent, and how to achieve this goal in a shop floor environment is the major concern here. Aspects to be considered for intelligence include autonomy, human interaction, change/failure recovery, quality control, resource management, high speed machining, and learning.

Autonomy. The software components organising the intelligent CNC controller should be able to operate autonomously without its activities being designated by the operator.

Minimised human decision and interaction. In a case when human intervention is necessary, it should be kept to a minimum, and the tasks to be handled by a human and those by a CNC controller should be clearly distinguished.

Change/failure recovery. The control system should be able to deal with unexpected change/failure occurring during the machining processes.

Quality control. To minimise the geometric error between the design part and machined part, on-machine measurement (OMM) of the part and proper diagnosis are required.

Adaptive control. For optimal machining, the cutting conditions should be adaptively controlled (instead of having constant values).

Resource management. All the resources involved in the machining process should be managed and controlled.

High-speed machining. The intelligent controller should support high-speed machining without the trade-off of surface quality deterioration.

Learning. The control system should be capable of acquiring machining knowhow and incorporating it into the knowledge base for the enhancement of the control system.

2.2 Data-Interface-Level Requirements

This data-interface level is associated with data interface scheme between CAM and CNC, and data manipulation within CNC. This level is crucial for overcoming the drawbacks of using conventional G-codes as an input, and for dealing with the limited data manipulation capability of a conventional CNC. A CAD data interface with a standard schema, internet interface, process planning, tool-path generation, and virtual machining are to be provided.

Direct interface with CAD data. To use the geometric meaning that is usually lost in the CAD-CAM-CNC chain, CNC should have the capability of a direct CAD data interface.

Use of standard data. There are international standards that can be applied to this area of manufacturing, such as ISO 10303 for product model data representation and ISO 13399 for representing small cutting tools. These should be reflected in the data-interface between CAM and CNC.

Internet interface. The control software should provide the means for an Internet interface for running and monitoring the machine tools at a remote site via the internet.

Seamless data interface. Often a part program changed on the shop floor, is different from that in the CAD/CAM system. To avoid this situation, a mechanism for bi-directional transfer of data between CAM and CNC is required.

Incorporation of process planning and tool-path generation. For intelligent control, CNC should be able to generate the tool path based on the process planning information.

Feature-based input. To generate the tool path autonomously, the input to the CNC controller should be a feature-based workpiece description.

Virtual machining. Before executing the actual machining task, the CNC should be able to operate in advance to enable any possible errors to be found.

2.3 Implementation-Level Requirements

Implementation-level requirements are concerned with the structural and topological relationship of the software modules and the implementation methodology. The methodology used by OSACA, OMAC, and OSEC should be considered. OSACA, OMAC, and OSEC are the representative consortia that have been engaged in developing the open architecture of CNC controllers in Europe, USA, and Japan, respectively. Software-based CNC, open and modular architecture, and user configurable structure are the major concerns here.

Software-based CNC. The most fundamental requirement is that all the CNC modules should be implementable purely by software under the most widely used operating system within a PC platform. An interface board should be the only hardware media between the CNC and the hardware (driver, motor, machine tools).

Open and modular architecture. CNC should be of open architecture, enabling the user to access the internal functions of CNC for any modifications. In order to operate in a plug-and-play fashion, it should be developed with a modular structure.

User configurability. The user should be able to reconfigure (edit) the modules of the CNC so that it can be adapted effectively to customer needs.

3. Functional Architecture

The basic architecture of CNC is composed of:

1. Human–machine interface (HMI).
2. Control functions.
3. Data processing within CNC.

This holds true for intelligent CNC. In the architecture presented, the user requirements are implemented in basic architecture in an extended fashion. As shown in Fig. 1, it is composed of:

1. *SFP/TPG* (shop floor programming/tool path generation) modules, which are extended HMI comprehensively covering part programming and tool path generation based on a STEP-NC data model, reflecting the *data-interface level requirements*,
2. *Control modules* covering various intelligent control functions, reflecting the *function-level requirements*,
3. *Common DB modules* providing comprehensive data for the SFP/TPG and control modules.

3.1 Control Modules

Control modules reflect the function-level requirements of CNC based on the holonic paradigm characterised by decision making, executing, and monitoring. The control modules, *Decision Maker*, *Executor*, *NCK/PLC*, *Monitor*, *Emergency Handler,* involve intra-task management of the CNC, and *Communicator* for the inter-task management (see Fig. 1). Further, modules for supporting non-machining functions, such as *Setup Manager*, *Inspector*, and *Learner* are included in the control modules.

Setup Manager. This supports the part set-up operation. Once the part is loaded on the machine, it finds the datum position by moving a touch probe using the workpiece and fixture geometry information.

Decision Maker. This schedules the task, selecting the next task from various alternatives out of a *nonlinear process plan*. The nonlinear process plan includes alternative process plans, and it can be represented by an AND-OR type graph to be explained later. One of the critical decisions is assigning the priorities between the scheduled task and the newly invoked task by the emergency handler and the inspector.

Executor. This converts the task into commands and passes them to NCK/PLC. If the task is a machining operation, it retrieves the corresponding tool path from the *Tool-Path DB* and passes it to NCK/PLC. If the task is a tool change, it finds the tool in the tool magazine and passes it to NCK/PLC. Executor keeps track of the commands executed by NCK/PLC for adaptive control.

NCK/PLC. NCK interprets the tool path commands and executes them by activating the servo mechanism, and PLC executes machinery commands, such as tool change and workpiece loading/unloading. For freeform surface machining, NCK is capable of NURBS interpolation in which accurate and high speed machining can be carried out with reduced data.

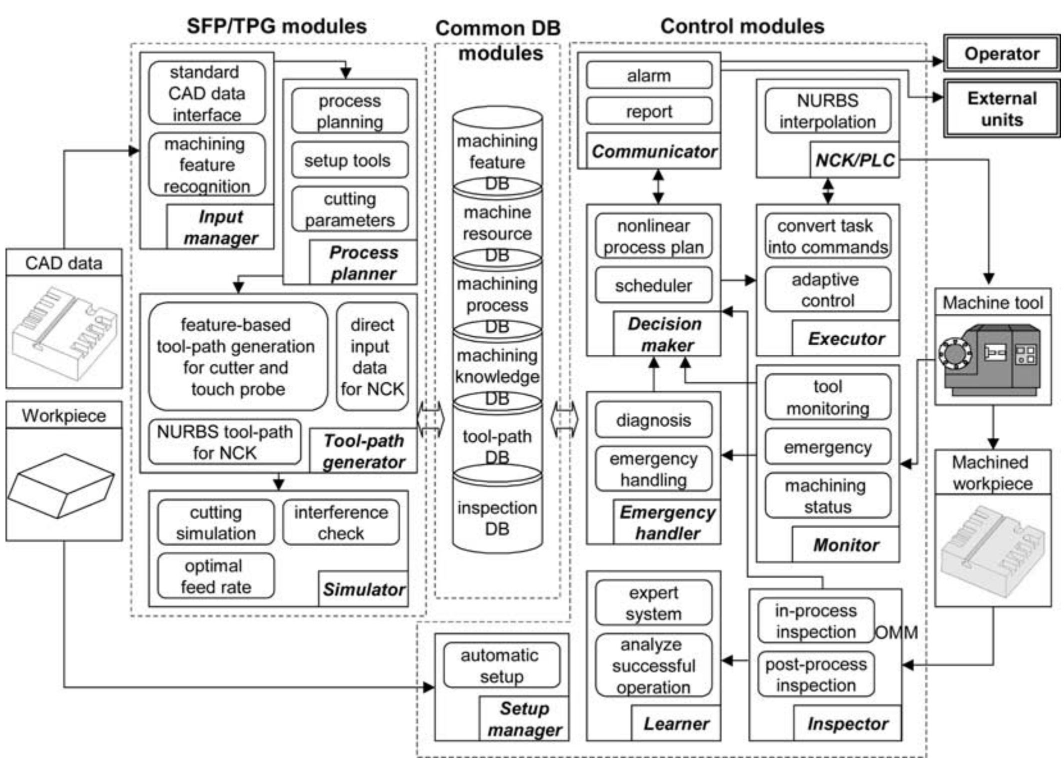


Fig. 1. A functional architecture of intelligent CNC.

Monitor. The entire machining status is continuously monitored by capturing information from sensor signals. Tool monitoring and emergency detection are crucial tasks. The results are sent to the emergency handler and/or the decision maker accordingly.

Emergency Handler. In case of an emergency, which is monitored and reported by the monitor, the emergency handler makes a diagnosis and decides what to do about it. The result is sent to the decision maker for the final decision and scheduling. For example, in the case of tool breakage, the emergency handler retracts the tool, and checks if an alternative tool is available in the tool magazine (through *Machine Resource DB*). If one is available the operation is resumed with the alternative tool, otherwise it reports to the decision maker and waits for a final decision. The emergency handler can be thought of as a subtype of the decision maker, specialising in dealing with emergency.

Inspector. In-process and post-process inspections are carried out automatically by the inspector. In either case, inspection is done on the machine tool by *OMM* (on-machine-measurement). The inspector generates the tool-path for the touch probe and stores the data into the *Inspection DB*. Any geometrical errors between the designed part and the machined part are found by comparing the data of the inspection DB with that of the *Machining Feature DB*.

Learner. Information captured during machining is analysed by an expert algorithm, and stored in the *Machining Knowledge DB*.

Communicator. The communicator is responsible for the interactions with external units, such as the CAD/CAM system, shop floor control system, and human operator:

1. When requested by the CAD/CAM system, the CNC sends the part program in the current CNC DB.
2. When requested by the shop floor control system, it reports the current status including the progress of machining, and problems that occurred during the machining.
3. When the execution of a certain operation is impossible due to unexpected problems it sounds an alarm for operator attention.

3.2 SFP/TPG Modules

The SFP/TPG modules reflect the data-interface level requirements of CNC. The SFP/TPG modules incorporate the CAM functions into the shop-floor programming system based on the STEP-NC data model. It includes *Input Manager*, *Process Planner*, *Tool-Path Generator*, and *Simulator*.

Input Manager. The roles of the input manager are CAD data interface and machining feature recognition. It translates the standard CAD data (STEP, AP203) into built-in geometric modelling kernel data, recognises the machining features, and extracts the feature attributes required for machining. Output is stored in the *Machining Feature DB*.

Process Planner. It determines the processing sequence, operations, fixtures, set-ups and cutting tools required to machine the features. The processing sequence is represented by a nonlinear process plan so that the decision maker can select an appropriate plan at the time of execution. Optimal cutting parameters, machining strategies and tools for operations are determined using the *Machining Knowledge DB*. For this, a knowledge-based process planning system is required. Output is stored in the *Machining Process DB*.

Tool-Path Generator. It generates tool paths both for machining and measurement. It can generate a complete path including approach, departure, and connection path between the machining or measurement paths. The generated tool paths are stored in the *Tool-Path DB*, which are accessed by NCK/PLC. As NCK/PLC is able to interpret NURBS curve directly, the tool-path generator does not segment the tool path of a freeform curve into lines/arcs.

Simulator. Prior to actual machining, it is required to perform a cutting simulation to verify the given tool path and to detect any possible errors. The simulator finds undercut or gouging and tool interference by cutting simulation. In addition to the error detection in the tool path, optimal feedrate is calculated by using the required material removal rate during the solid cutting simulation. Output is stored in the *Tool-Path DB* and the *Machining Process DB*.

3.3 Common DB Modules

Common DB modules are the repositories of data that are generated, updated, and retrieved by control modules and SFP/TPG modules. Machining feature DB, machining process DB, tool-path DB, and inspection DB are short-term databases and machine resource DB and machining knowledge DB are long-term databases. On completion of the part machining, short-term database are cleared.

Machining Feature DB. This stores machining feature information generated by the input manager. Stored data is the feature-based input for the process planner.

Machine Resource DB. This stores data on machine configuration, available tools, tool magazine, jigs/fixtures and sensors. It is updated by the operator or decision maker.

Machining Process DB. This stores nonlinear process plans generated by the process planner. A machining process is described by machining feature and machining operation, including machining strategy, cutting conditions and tools.

Machining Knowledge DB. This stores long-term machining knowledge that is used by the process planner and executor in collaboration with the expert system. It is updated by learner.

Tool-Path DB: This stores tool paths generated by the tool-path generator. It is accessed by the NCK/PLC and simulator.

Inspection DB. It stores tool paths for inspection and the inspected results generated by inspector.

4. Data Model for Intelligent CNC

To implement the intelligent CNC, various technologies should be developed. Among them, the data model is the key. A conventional data model mainly specifying the axis motion of the machine tool by M & G codes (ISO 6983) cannot provide the information required for the intelligent CNC. For such a purpose, a new data model formalised as ISO 14649 should be used. ISO 14649 often called STEP-NC is being developed by ISO TC 184 SC1, and it is currently available as a DIS-ballot version. Although its details may be changed a little, its structure will be maintained.

ISO 14649 is a structured data model for representing the process plan and execution strategy for NC machining based on the STEP (ISO 10303) paradigm. As illustrated in Fig. 2, it contains comprehensive information including, process (*workingstep*) sequence, manufacturing feature, machining operation, machining strategy, cutting tool geometry. These can be effectively applied in a *basic* type of STEP-compliant CNC. For *intelligent* control, the following data models should be provided [3].

4.1 Data Model for Nonlinear Process Plan

Typically, the process plan is given in a linear order in conventional CNC. A process plan given in this way does not provide room for the flexibility required for CNC. Suppose that tens of holes are to be machined. It would be better to declare that they should be machined in an unordered serial fashion than to specify a fixed sequence in a linear fashion. To cope with unexpected changes on the shop floor and/or to optimise the process sequence, a nonlinear process plan (NLP) including alternative plans should be given to the CNC so that the most appropriate one can be chosen by the CNC. For such a purpose, a data model for a nonlinear process plan is required for intelligent control.

In the architecture presented (Fig. 1), a nonlinear process plan is formed by the process planner in the SFP/TPG module and is used by the decision maker to select the next task based on the selection algorithm. A nonlinear process plan can be represented by a graph, where a node represents the machining process and an edge represents the relationship between the nodes. There can be seven types of relationship of execution in NLP:

1. *Sequential*: execution by a sequence.
2. *Conditional choice*: binary choice according to the given condition.
3. *Iterative*: repetitive execution.
4. *Concurrent*: all machining processes are started at the same time.

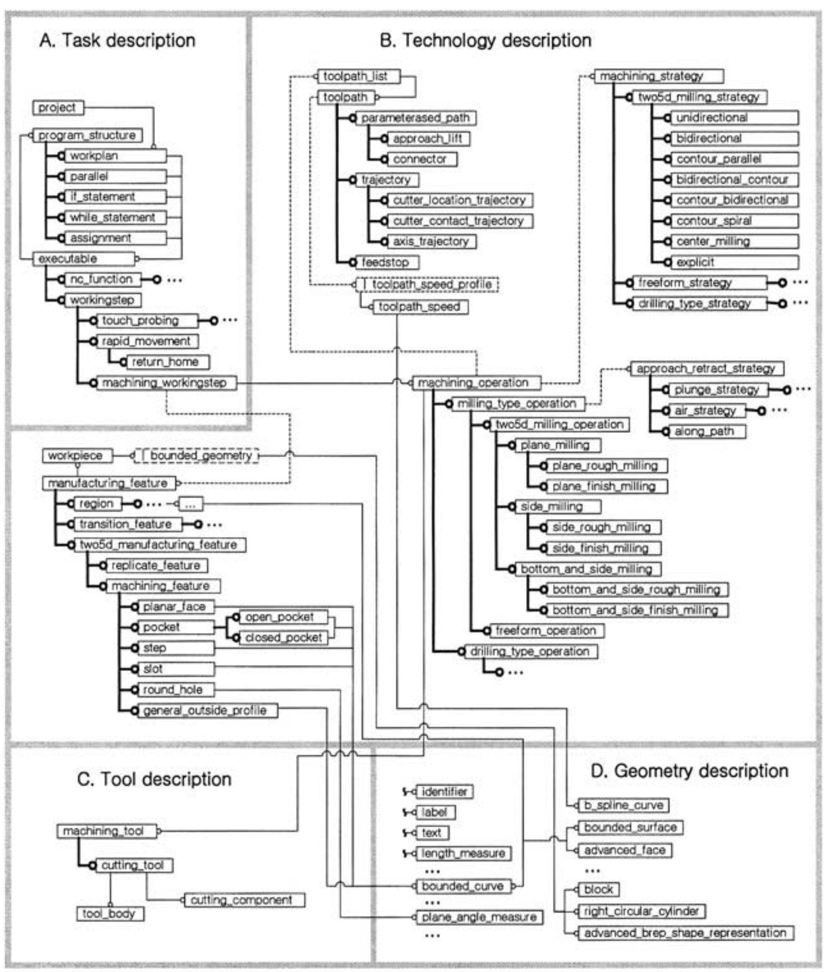


Fig. 2. The overall data structure of ISO 14649.

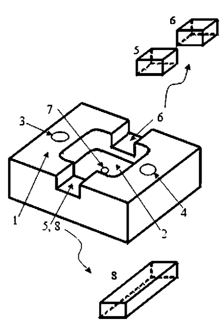
5. *Unordered serial*: all machining processes have to be executed irrespective of order.
6. *Selective*: execution of only one among several machining processes.
7. *Parallel*: all machining processes do not need to be started simultaneously.

A data model incorporating the above is given in Fig. A1 in the Appendix. Figure 3(*c*) shows an example of a nonlinear process plan represented by an AND-OR graph for the part given in Fig. 3(*a*). Figure 3(*b*) is a process sheet described by the terminology defined in ISO 14649. The process sequence graph (Fig. 3(*c*)) includes a total of 42 sequences including {1, 3, 4, 8, 2, 7} and {1, 2, 5, 6, 7, 3, 4}.

4.2 Other Data Models for Intelligent Control

Geometric information of clamping device. It is necessary to check if the cutting tool interferes with other structural entities such as clamping devices. Similarly, to avoid the collision of a cutting tool with a clamp, the geometric information of the clamp should be provided for the intelligent CNC controller. The tool-path generator, simulator, setup manager, and inspector modules use this information to generate an error-free path

(*a*)



(*b*)

WORKINGSTEP		
ID	FEATURE	OPERATION
1	planar_face	plane_rough_milling
2	closed_pocket	bottom_and_side_rough_milling
3	round_hole	drilling
4	round_hole	drilling
5	slot	bottom_and_side_rough_milling
6	slot	bottom_and_side_rough_milling
7	round_hole	drilling
8	slot	bottom_and_side_rough_milling

(*c*)

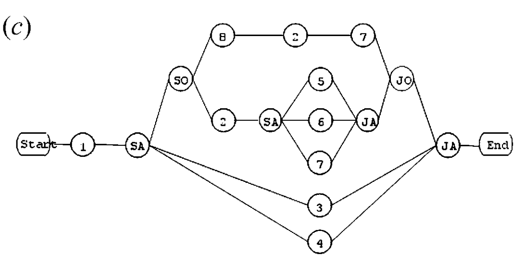


Fig. 3. The concept of the nonlinear process plan. (*a*) A part to be machined. (*b*) The process sheet. (*c*) The process sequence graph.

for the cutting tool and the touch probe. A schema for this is given in Fig. A2.

Data model for automatic set-up. An automatic set-up method is required for intelligent control. Along with the geometry of the workpiece, the description of the motion for automatic part set-up is required. Data relevant to automatic set-up is the geometry of the workpiece, information on the indexing table on which wokpiece is counted. This is required for the setup manager. Figure A3 shows a schema for this.

Data model for on-machine measurement (OMM). To inspect the geometric accuracy of the machined part OMM is necessary. Inspection should be treated as a *workingstep*. Depending on the time to be inspected, it could be either in-line or post-line process. A data model for the target area to be inspected is necessary. This is required for the A4 inspector module. A schema for this is shown in Fig. A4.

Data model for tool monitoring. It is necessary to monitor the status (wear and breakage) of the cutting tool. It should reflect the monitoring methodology including the target of monitoring (such as breakage, wear), types of sensor to be used (such as cutting force sensor, feed force sensor, spindle motor sensor, proximity sensor, acoustic emission sensor) (e.g. [13]), and the detection algorithm. This is required for the monitor module. A schema for this is shown in Fig. A5.

Data model for adaptive control. This is required for optimal machining which can be achieved by variable cutting conditions (instead of constant values). As in tool monitoring, the data model for adaptive control should reflect the adaptive control methodology including the target of control (such as feedrate or spindle speed), types of sensor to be used (e.g. [14]). This is required for the executor module. A schema for this is shown in Fig. A6.

5. Operational Scenario

Assuming that the presented intelligent CNC and data model (ISO 14649) are developed, an operational scenario is given to illustrate how it works. Depending on the degree of sophistication, different scenarios can be developed. The following scenario is realistic based on interactive operation (rather than fully automated). The scenario is divided into two parts:

1. The part programming stage,
2. The CNC operation stage.

5.1 Part Programming with SFP

The part programmer (user) designs a part (e.g. Fig. 3(*a*)) to be machined as a workpiece in a CAD system supporting an AP 203 data model (Fig. 4). Then, the user goes to a shop floor programming (SFP) system installed in either an off-line CAM system (external SFP) or a CNC system (built-in SFP). In the SFP system, the user is asked to specify the AP 203 files by the input manager module. Then, the input manager recognises the machining features and stores them in the machining feature DB. For each machining feature, a process plan is specified in the process planner module in terms of

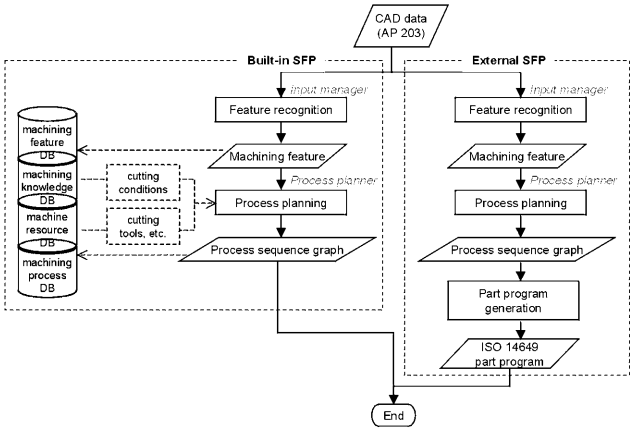


Fig. 4. The part programming scenario with the SFP system.

workingstep including machining operation and strategy together with cutting tools and cutting conditions specified in the process planner module.

Considering the shape of the machining features, the user provides an alternative sequence of *workingsteps* graphically, as shown in Fig. 3(*c*). When asked to input cutting conditions, the user specifies reference values and uses the adaptive control mode. However, the user does not specify the tool path for each *workingstep* as the intelligent CNC will do that. Also, the user specifies inspection tasks on the process sequence graph. However, since the cutting tools available in the tool magazine, and the monitoring and adaptive control schemes were previously given in the machine resource and machining knowledge DBs, they are not specified. Once the input is given by the user, SFP can automatically generate the part program in ISO 14649 format. However, if built-in SFP is used, this is not necessary, since all the input information is stored in the CNC DBs.

5.2 CNC Operation Stage

If the part program is provided externally, it is downloaded to CNC via the Internet or DNC line. Then, CNC translates and stores the information contents in the appropriate DBs. If the part program is prepared with SFP built in CNC, this step is not necessary (Fig. 5). CNC generates the tool path for the cutter and touch probe (by tool-path generator), which can be shown graphically by the simulator. The computed tool path will be used as a reference tool path during the machining operation. Note that the tool path may be generated during the machining operation. To alleviate the computational burden, the former is preferable.

After loading the workpiece, the operation starts by pressing the cycle start button. The operation is initiated by finding the datum position automatically using a touch probe (by *setup manager*). Then, based on the process sequence graph (representing the nonlinear process plan), the decision maker determines the next *workingstep* to be executed. If all the resources (e.g. cutting tool) specified are available (resp. in the tool magazine), it orders the executor to execute the *workingstep*. Then, the executor accesses the tool-path DB and converts it into commands for NCK/PLC and invokes the adaptive control mechanism to provide variable cutting conditions. This operation is continuously monitored by the monitor.

When a tool breakage is detected, it stops the operation and invokes the emergency handling mechanism, followed by reporting to the *decision maker*. The *decision maker* checks if alternative cutting tools (the tool size could be different from the broken tool) are available in the tool magazine (via machine resource DB). If available, it orders the tool-path generator to generate a new tool path for the substituted cutting tool to remove the remaining volume. Otherwise, it reports to the *communicator* that the *workingstep* is inexecutable. Then, the *communicator* warns the human operator to supply a new cutting tool. In the meantime, the decision maker tries to find an alternative sequence based on the nonlinear process plan. If one is found, it determines the next *workingstep*, and informs the tool-path generator. The tool-path generator, checks the *validity* of the tool path (in the tool-path DB) for the working step, and may generate a new one if necessary.

When the inspection *workingstep* is required, the decision maker orders the inspector to invoke the necessary action: measuring the target area, storing the measured data in the inspector DB. Then it computes the geometric error by comparing the measured data with the machining-feature DB. If it is out of tolerance, it reports to the *decision maker* that remachining is required. Then, a new *workingstep* for remachining is included by the *decision maker*. Upon completion of a *workingstep*, the decision maker updates the process sequence graph (machining process DB), state of machine resource (machine resource DB), and machining knowledge DB based on the results produced during the operation. Then, the next *workingstep* is determined. This process is repeated over and over until all the *workingsteps* are completed. At any time when requested by the shop floor control system (resp. CAD/CAM system), the decision maker orders the *communicator* to send the current status of the execution, and progress (the current part program) based on the contents of the DBs.

6. Concluding Remarks

In this paper, a conceptual framework for an intelligent CNC is presented in terms of a functional architecture. It is based on the perspective of a shop floor control system, and requirement analysis in three levels:

1. Functional-level.
2. Data-interface level.
3. Implementation-level.

The architecture is composed of:

1. SFP/TPG modules.
2. Control modules.
3. Common DB modules.

We suggest that the STEP-NC data model should be used for developing such modules. Also, for the purposes of intelligent control, we point out that the STEP-NC data model must be

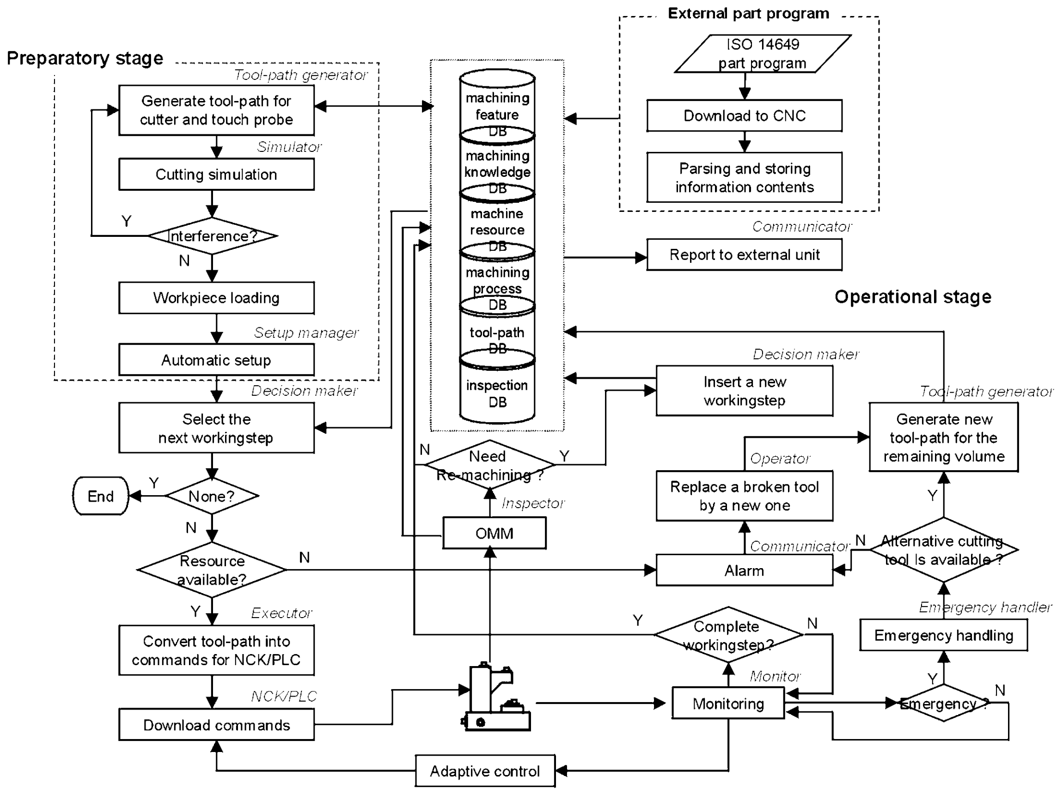


Fig. 5. The operation scenario in SNC.

updated by incorporating data models for the nonlinear process plan. Assuming these data models are available, an operational scenario is given to illustrate how the presented CNC works. Through the scenario, it is shown that CNC is capable of executing the assigned task intelligently while dealing with disturbances autonomously. The framework presented can be used as a paradigm for a STEP-compliant CNC.

The CNC presented can be implemented on a PC, purely by software interfaced with hardware via an interface board. Depending on the modules, the degree of requirement for real-time operation varies. In general, the control modules deal with time-critical tasks, but non-real-time operation may be acceptable for the SFP/TPG modules. It may require dual CPUs for implementing the two groups of modules. Even in such a case, the amount of computing should be taken into consideration in developing algorithms for the modules. Simple and robust algorithms will be a crucial factor for the implementation. Currently, we are developing an intelligent CNC called ASNC (autonomous STEP-compliant CNC) based on the scheme presented. More details will be given in a future paper.

Acknowledgements

This research is in part supported by a grant for the National Research Laboratory for STEP-NC Technology by MOST (Ministry of Science and Technology) in Korea, and grant (no. 1999–2-315–002–3) for the interdisciplinary research program of KOSEF (Korea Science and Engineering Foundation).

References

1. ISO TC184/SC1/WG7, “ISO 14649: Data model for computerized numerical controllers” (DIS ballot version), 2000.
2. J. H. Cho, “STEP-NC architecture design and on-line tool path generation for autonomous control”, PhD dissertation, POSTECH, 2001.
3. S. H. Suh, “Propostions for future update of ISO 14649 based on DIS-ballot version”, ISO TC184/SC4 meeting, Funchal, Portugal, 20 February 2001.
4. J. P. Kruth, “A prototype NC controller driven by feature-based part description”, Proceedings of the 4th Pacific Conference on Manufacturing, Jakarta, Indonesia, pp. 535–544, 1994.
5. S. H. Suh, H. D. Hong and J. H. Cho, “Feature-based CNC control and message specification supporting HMS paradigm”, CIRP Journal of Manufacturing Systems, 29(5), pp. 437–441, 1999.
6. H. Brussel, “Holonic manufacturing system, the vision matching the problem”, Proceedings of First European Conference on Holonic Manufacturing Systems, Hanover, 1994.
7. L. Bongaerts, P. Valckenaers, H. Brussel and J. Wyns, “Schedule execution for a holonic shop floor control system”, Preprints of the ASI-95 of NOE on ICIMS, Lisboa, pp. 115–124, 1995.
8. L. Bongaerts, P. Valkernaers, H. Brussel and P. Peters, “Schedule excution in holonic manufacturing system”, Proceedings of the 29th CIRP International Seminar on Manufacturing Systems, Osaka University, Japan, pp. 209–215, 1997.
9. K. Yamazaki, Y. Hanaki, M. Mori and K. Tezuka, “Autonomously proficient CNC controller for high-performance machine tools based on an open architecture concept”, Annals CIRP, 46(1), pp. 275–278, 1997.
10. N. Brouer and M. Weck, “Feature-oriented programming interface of an autonomous production cell”, Proceedings of the 4th IFAC

workshop CIRP International Seminar on Intelligent Manufacturing Systems, Seoul, Korea, pp. 223–228, 1997.
11. J. Liu and K. Yamajaki, "Agile production realization based on autonomously proficient CNC controller infrastructure," Proc. of the 31st CIRP Int'l Seminar on Manufacturing Systems, Berkeley, CA, USA, pp. 414–419, May 26–28, 1998.
12. G. Pritschow, C. Daniel, G. Junghans and W. Sperling, "Open system controllers – a challenge for the future of the machine tool industry", Annals CIRP, 42(1), pp. 449–452, 1993.
13. J. Tlusty and G. C. Andrews, "A critical review of sensors for unmanned machining", Annals CIRP, 32(2), pp. 563–572, 1983.
14. J. H. Tarn and M. Tomizuka, "On-line monitoring of tool and cutting conditions in milling", Journal of Engineering for Industry, 111, pp 206–212, 1989.

Appendix

```
ENTITY relationship
  ABSTRACT SUPERTYPE OF (ONE OF (
    sequential, unordered_serial,
    conditional_choice, selective,
    iterative, concurrent, parallel)
END_ENTITY ;

ENTITY sequential
  SUBTYPE OF (relationship) ;
  its_elements :  LIST [0:?] OF process ;
END_ENTITY ;

ENTITY unordered_serial
  SUBTYPE OF (relationship) ;
  its_elements :  SET [0:?] OF process ;
END_ENTITY ;

ENTITY conditional_choice
  SUBTYPE OF (relationship) ;
  true_element :  process;
  false_element :  process;
END_ENTITY ;

ENTITY selective
  SUBTYPE OF (relationship) ;
  its_elements :  SET [0:?] OF process ;
END_ENTITY ;

ENTITY iterative
  SUBTYPE OF (relationship) ;
  its_elements :  LIST [0:?] OF process ;
END_ENTITY ;

ENTITY concurrent
  SUBTYPE OF (relationship) ;
  its_elements :  SET [0:?] OF process ;
END_ENTITY ;

ENTITY parallel
  SUBTYPE OF (relationship) ;
  its_elements :  SET [0:?] OF process ;
END_ENTITY ;
```

Fig. A.1. An EXPRESS schema of nonlinear process plan.

```
ENTITY clamp
  its_geometry :     advanced_brep_shape_representation ;
END_ENTITY ;

ENTITY workpiece
  ...
  its_clamp :            clamp ;
  clamping_position :    SET [0:?] OF cartesian_point ;
  ...
END_ENTITY ;
```

Fig. A.2 An EXPRESS schema of a clamp.

```
ENTITY setup
  its_origin :              axis2_placement_3d ;
  its_secplane :            elementary_surface ;
  its_workpiece_position :  LIST [0:?] OF workpiece_position ;
  its_index_table :         index_table ;
  its_probing :             touch_probing ;
END_ENTITY ;
```

Fig. A.3 An EXPRESS schema of automatic setup.

```
TYPE OMM_type = ENUMERATION OF (
  in_process_measurement, post_process_measurement) ;
END_TYPE ;

ENTITY OMM
  its_type :              OMM_type ;
  measurement_path :      tool_path ;
  measured_results :      user_defined_database ;
  desired_tolerance :     toleranced_length_measure ;
  design_surface :        advanced_brep_shape_representation ;
END_ENTITY ;
```

Fig. A.4 An EXPRESS schema of OMM.

```
ENTITY monitoring
  ABSTRACT SUPERTYPE OF (monitoring) ;
END_ENTITY ;

TYPE monitoring_sensor_type = ENUMERATION OF (
  dimensional_sensor, cutting_force_sensor, feed_force_sensor,
  spindle_motor_sensor, acoustic_emission_sensor) ;
END_TYPE ;

TYPE monitoring_target_type = ENUMERATION OF (
  dimensional_check, tool_wear, tool_breakage) ;
END_TYPE ;

ENTITY tool_monitoring
  SUBTYPE OF (monitoring) ;
  its_sensor :     monitoring_sensor_type ;
  its_target :     monitoring_target_type ;
END_ENTITY ;
```

Fig. A.5 An EXPRESS schema of tool monitoring.

```
ENTITY adaptive_control
  its_sensor :               monitoring_sensor_type ;
  max_torque :               OPTIONAL REAL ;
  max_cutting_force :        OPTIONAL REAL ;
  its_control_algorithm :    OPTIONAL adaptive_control_algorithm ;
END_ENTITY ;

FUNCTION adaptive_control_algorithm
  ......
END_FUNCTION ;
```

Fig. A.6 An EXPRESS schema of adaptive control.