Int J Adv Manuf Technol (2013) 68:2791–2800
DOI 10.1007/s00170-013-4871-7

ORIGINAL ARTICLE

Five-axis STEP-NC controller for machining of surfaces

Hongbin Liang · Xia Li

Received: 30 September 2012 / Accepted: 25 February 2013 / Published online: 6 April 2013

Abstract Five-axis machining is more widely used in manufacturing of freeform surfaces. However, in five-axis machining of freeform surfaces, incomplete information exchange between computer numerical control (CNC) and computer-aided design/computer-aided manufacturing (CAM) results in many limitations need to be rectified. In the paper, a new structure of CNC based on STEP-NC standard is proposed, where tool path planning, tool offset, and inverse kinematics are transferred from CAM to CNC. In order to guarantee good openness, open platform and standard interface are applied in the development. Technology of module collaboration and design of data flow are studied. A five-axis real-time interpolator for non-uniform rational B-spline surfaces machining is realized. Based on these technologies, a five-axis CNC is developed in the manner of software realization, which consists of interpreter, task coordinator, axis group, softPLC, etc. The software CNC system has been applied on a tilt-rotary type five-axis machine tool, where the milling experiment has been performed successfully.

Keywords Five axis · STEP-NC · Real-time interpolator · Free-form surface

1 Introduction

In recent decades, five-axis machining is more widely used in the manufacturing of freeform surfaces to replace three-axis machining due to its benefits on high machining efficiency and fine surface finish. Generally, a typical procedure for five-axis machining freeform surfaces includes three parts. Modeling of surface is done in computer-aided design (CAD), while path planning, tool offset, and postprocessor are done in computer-aided manufacturing (CAM), real-time interpolation in computer numerical control (CNC) systems [1–3].

Due to the broad range and high complexity of technology concerned about five-axis machining of freeform surfaces, in the paper, the main research efforts in the field of interpolation technology and numerical control (NC) programming interface standard are selected to analyze as follows.

1.1 Research on interpolation technology

Conventional CNC machine tools generally provide only functions for linear and circular interpolations. During the machining of freeform surfaces, surfaces have to be approximated offline forming series of piecewise line in the CAM system using allowable tolerance assigned by customers. This approximation results in unavoidable errors between the desired paths and the commanded paths.

To overcome these problems, a variety of spline interpolators have been investigated worldwide since the 1980s [4, 5]. Among them, many achievements in non-uniform rational B-spline (NURBS) curve interpolation have been obtained [6, 7]. Zhang and Greenway developed a Taylor's first-order expansion algorithm [8]. Cheng proposed a Taylor's second-order expansion algorithm [9]. Park realized a two-stage interpolator [10]. Authors also focus the research on the NURBS curve interpolator and proposed a NURBS curve interpolator for adaptive feed speed [11].

Although NURBS curve interpolators have been shown to be effective, there are some problems to be solved. The method of the decomposition of a surface into curves leads to the information loss of normal vectors of cutter contact (CC) points on surface to be machined and tangent vectors of cutting path. Due to information loss, 3D cutter compensation cannot

H. Liang (✉)
College of Mechanical Engineering, University of Science and Technology Liaoning, Anshan, Liaoning Province 114044, China
e-mail: lianghb@hit.edu.cn
URL: http://www.ustl.edu.cn

X. Li
College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin, Heilongjiang Province 150001, China

be realized online in CNC. In addition, due to dividing of the functions of path planning, cutter offset and real-time interpolation into two parts, i.e., CAM and CNC, when size of tool changes, even very little, programmer has to return to generate a new part program in CAM, then passes the modified program to CNC again to machine work. It can be seen that this way results in productivity reduction.

To solve these problems, the attempt at the surface interpolators has started since the 1990s in the last century, but most of work is not related to the NURBBS surface interpolators. Cheng Chung-Wei proposed a NURBS surface interpolator, which focused on three-axis machining [12].

1.2 The research on NC programming interface

In order to avoid information loss on exchanges between CAD/CAM and CNC, Canadian Professor Altintas developed a system integrated CAD information to machine freeform surface [13]; Professor Jeard of New Hampshire University in the USA studied Numerical Control Marked Language language to replace G code (ISO 6983) [14]. Although these methods proved effective in some aspect, they still had limitations on acceptance and perfection.

In the 1990s, under the joint efforts worldwide, STEP-NC interface was released as ISO standard, which aims at replacing ISO 6983 (G Code). Its benefits have been addressed by numerous researchers in literatures [15–17].

STEP-NC supports the description of machining freeform curves and surfaces, where surfaces and curves are described in the form of NURBS. However, as claimed above, currently most commercial CNC systems have not provided interpolation of NURBS surface. In the field of machining surfaces based on STEP-NC, STEP manufacturing group led by Professor Martin is a very active international research team. In the demonstrations of their outcome, machined work parts with surfaces were generally selected [18]. In the method, in order to research online cutter compensation, cutter location (CL) data independent on machine configuration were included in NC part program written in STEP-NC. However, the approach is only suitable for little worn cutter compensation. Although geometric information of surface and the orientation of cutter axis are available in NC program, CC data corresponding to CL data are not available, where the method is difficult to calculate compensation vector. And path intervals also need to modify, especially in the case of tool changing and tool size varying. Finally, it cannot be avoided that path planning would be done again for reprogramming in CAM. Similarly, this way has not made full use of the benefits of STEP-NC, so it is only a temporary compromised approach. It should be noted that STEP-NC standard are in the initial stage of research and application, so the efforts on five-axis real-time interpolation for NURBS surfaces machining are still relatively less.

1.3 The proposed method in the paper

Just as discussed above, the dividing of manufacturing system into several parts, such as CAD, CAM, and CNC, etc., has lead to the difficulty in instantaneous and complete information exchange among these systems. Authors think the weak computational power of computers at that time is one of the main reasons. With the advance of computer technology and the progress of CNC technology, CNC can access to data of all manufacturing resources at factory floor especially after using new programming interface standard STEP-NC so that many tasks that are performed currently in CAM would be done real-time in CNC in the future.

Authors believe that CAD, CAM, and CNC should be taken as an integral part to be considered instead of individuals in the research of five-axis NURBS surfaces machining. So, firstly a new structure of CNC system has been proposed (see Fig. 1) in the paper, where path planning, tool offset and inverse kinematics are transferred from CAM to CNC, and STEP-NC are used as the programming interface for five-axis machining of NURBS surface. Secondly, five-axis interpolation of NURBS surface based on STEP-NC has been studied. Thirdly, one open five-axis CNC system has been developed. Finally, the milling experiment has been performed.

2 Five-axis real-time interpolator for NURBS surfaces based on STEP-NC

There are no instructions to define NURBS surfaces interpolation in G Code, researchers have to define and use proprietary instruction format in the past. When new CNC adopts STEP-NC as part program interface, the standard instruction format in STEP-NC should be provided to ensure complete information be transmitted to CNC, including geometric information of work part and technological data.

2.1 Description of NURBS surfaces and technological data in STEP-NC standard

A freeform surface in 3D space can be expressed as:

$$S(u,v) = x(u,v)\ \overrightarrow{i} + y(u,v)\ \overrightarrow{j} + z(u,v)\ \overrightarrow{k}\ \ u,v \in [0,1] \quad (1)$$

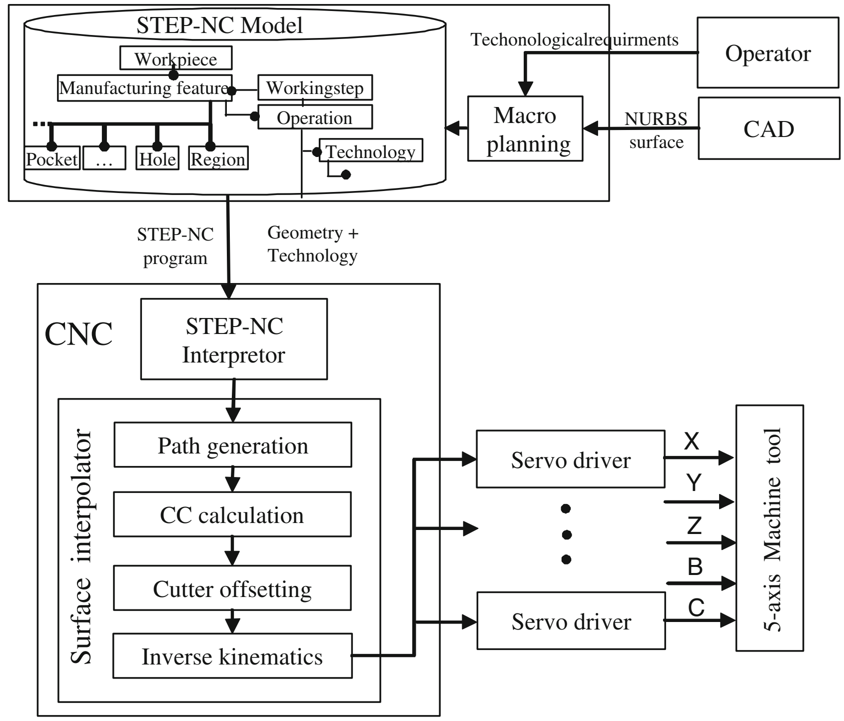


Fig. 1 Proposed method for the machining of surfaces

The corresponding NURBS surface can be expressed as:

$$S(u,v)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)W_{i,j}C_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)W_{i,j}} \tag{2}$$

Where $C_{i,j}$ (i=0,1,…,n; j=0,1,…,m) are called control points; $W_{i,j}$ are called the corresponding weights of $C_{i,j}$; $N_{i,p}(u)$ and $N_{j,q}(v)$ are called pth degree and qth degree B-spline basis functions(blending functions), defined on the knot vectors,

$$\mathrm{U}=\{\underbrace{0,\cdots,0}_{p+1},u_{p+1},\cdots,u_{r-p-1},\underbrace{1,\cdots,1}_{p+1}\}$$
$$\mathrm{V}=\{\underbrace{0,\cdots,0}_{q+1},v_{q+1},\cdots,v_{s-q-1},\underbrace{1,\cdots,1}_{q+1}\}$$

Blending functions $N_{i,p}(u)$ and $N_{j,q}(v)$ can be defined as follows:

$$N_{i,1}(u)=\begin{cases}1(u_i\le u<u_{i+1})\\0(\text{otherwise})\end{cases}$$
$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p-1}-u_i}N_{i,k-1}(u)+\frac{u_{i+p}-u}{u_{i+p}-u_{i+1}}N_{i+1,p-1}(u) \tag{3}$$

$$N_{j,1}(v)=\begin{cases}1(v_j\le v<v_{j+1})\\0(\text{otherwise})\end{cases}$$
$$N_{j,q}(v)=\frac{v-v_j}{v_{j+q-1}-v_j}N_{j,q-1}(v)+\frac{v_{j+q}-v}{v_{j+q}-v_{j+1}}N_{j+1,q-1}(v) \tag{4}$$

According to the mathematical definitions, it can be seen that a NURBS surface can be determined by four kinds of parameters: the order of basic functions, control points, knot vectors, and weights.

Referring to the definition of B-spline surface in part 10 of ISO14649, it can been seen that some derived class from B-spline surface are also defined, including rational B-spline surface, uniform surface, quasi-uniform surface, Bezier surface, and B-spline surface with knots. But the defining each of these surfaces is not totally equivalent to the above mathematical definition of NURBS surface. It can be found that B-spline surface with knots includes three kinds of parameters: control points, knot vectors and the order of basic functions; rational B-spline surface includes the order of basic functions, knot vectors and weights; and their relationship is ANDOR. So referring to the definition of ANDOR and rules about how to name a new entity, using rational B-spline surface and B-spline surface with knots, authors derive a new entity—*b_spline_surface_with_knots_and_rational_b_spline_surface*. Its main attributes are represented clearly in Fig. 2, which shows that the entity of NURBS surface inherits the attributes of parents and the corresponding rules, including the order of basic functions, control points, knot vectors, and weights. The new entity is defined in EXPRESS language as follows:

```
ENTITY
  b_spline_surface_with_knots_and_rational_b_spline_surface
    SUPERTYPE OF (b_spline_surface_with_knots, rational_b_spline_surface);
 END_ENTITY;
```

Except geometric information, technological information is also defined in STEP-NC. For milling of a freeform surface, the technological requirements are defined in entity freeform_operation, including cutting tool, feed, approach and retract strategy, chordal tolerance, scallop height, etc. Some parameters related to five-axis machining such as inclined angle, tilted angle may also be specified by the entity freeform_operation.

Thus, according to STEP-NC standard, an NC program for machining surfaces may use the entity b_spline_surface_with_knots_and_rational_b_spline_surface to exactly describe the shape of part with NURBS surface, and use the entity freeform_operation to specify technological data. Consequently, both modeling information from CAD and technological information created in CAM may be inputted to a STEP-NC controller system.

2.2 Five-axis interpolator

The proposed interpolator for machining of NURBS surfaces is suitable for flat-bottom mills, which may obtain better machining quality and higher machining efficiency compared to ball-end mills. The interpolator must calculate real-time the position increments of five-axis in the set interpolation period, where the three main items should be included: (a) tool path planning based on isoparametric method, (b) calculation of cutter location point and orientation based on constant feed rate, and (c) inverse kinematics transformation independent of machine configuration.

2.2.1 Tool path planning

Path planning based on isoparameter is to decompose a NURBS surface into a group of isoparametric curves (CC paths), shown in Fig. 3, where parameter u or v keep constant on each curve. The increments of parameter u or v between a pair of adjacent curves should meet requirement that the maximum scallop height must be controlled within an allowable value. In the paper, we suppose first milling in u direction then milling in v direction. A cutter feeds along these curves to complete the milling the whole surface.

During the interpolation for the cth path $S(u, v_c)$, the value of parameter increment $\Delta v_{c,k}$ between the two corresponding points in the current path and the following CC path can be calculated from

$$\Delta v_{c,k} = \Delta l \Big/ \left(\left(\vec{n} \times \vec{t} \right) S_v(u_k, v_c) \right) \tag{5}$$

Where $\vec{t}$ and $\vec{n}$ are the surface unit tangent and the normal vectors respectively, at the interpolated point $S(u_k, v_c)$, and Δl is the incremental length of the CC path interval which is based on the allowable scallop height h, the cutter radius r, and the radius of curvature along path interval R_v.

During the interpolation of CC paths, using Eq. (5) obtains the corresponding $\Delta v_{c,k}$ on all of the interpolated points; choosing the minimum value of all $\Delta v_{c,k}$ as the

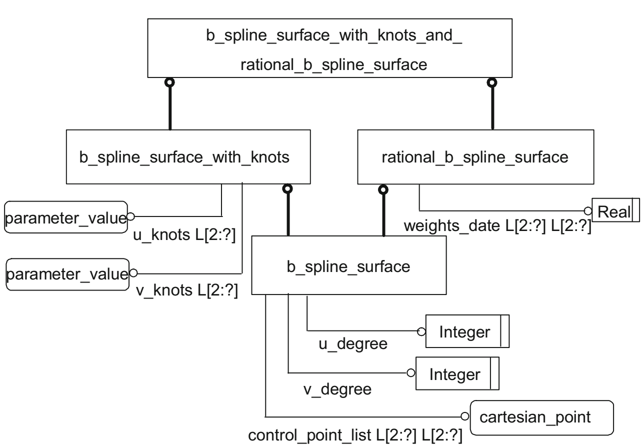


Fig. 2 The simplified EXPRESS-G for NURBS surface entity

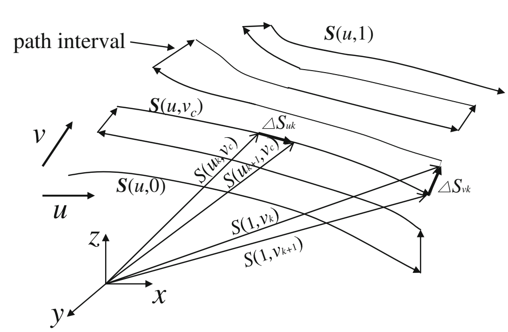


Fig. 3 Tool path planning and interpolation for NURBS surface

parameter increment to calculate the following CC path, it is also v-direction increment to determine the path interval.

2.2.2 Calculation of cutter location point and orientation based on constant feed rate

This job can be finished in following two steps: one is to calculate the coordinates of CC points, another is calculation of tool offset.

Calculation of CC points: Because the isoparametric approach is adopted, where the value of parameter v is fixed along one tool path and only the value of parameter u varies, the calculation of CC points is virtually converted to the interpolation of NURBS surfaces.

Supposed that cutter moves at the speed V_{cc} along the cth isoparametric tool path $S(u, v_c)$ on the NURBS surface $S(u, v)$, V_{cc} can be expressed as

$$V_{cc} = \left\| \frac{dS(u, v_c)}{dt} \right\| \tag{6}$$

Supposed interpolation period is T, the relationship between the value of u_k at the current interpolation time $t_k=kT$ and the value of u_{k+1} at the following interpolation time $t_{k+1}=(k+1)T$ can be expressed as

$$u_{k+1} = u_k + \frac{TV_{CC} + (T^2/2)(dV_{CC}/dt)}{\sqrt{(x')^2 + (y')^2 + (z')^2}} - \frac{(TV_{CC})^2 (x'x'' + y'y'' + z'z'')}{2\left((x')^2 + (y')^2 + (z')^2\right)^2} \tag{7}$$

Substituting the resultant (u_{k+1}, v_c) into Eq. (1), CC point can be obtained.

Tool offset: In order to machine parts, the data of CC point must be converted to the CL data. The CC data includes cutter's position vector C and the inclination angle λ and tilted angle ω; while CL data includes the cutter's center position vector P and orientation, also referred to as tool axis orientation vector O. The conversion procedure is called tool offset. Through geometry transformation, CL data are calculated:

$$P = CC + r\left(-\overrightarrow{f}\cos\lambda\cos\varpi + \overrightarrow{n}\sin\lambda - \overrightarrow{b}\cos\lambda\sin\varpi\right) \tag{8}$$

$$O = \overrightarrow{f}\sin\lambda\cos\omega + \overrightarrow{n}\cos\lambda + \overrightarrow{b}\sin\lambda\sin\omega \tag{9}$$

Where CC is the coordinate of CC point, r is the radius of the cutter, $\overrightarrow{f}$ is the unit vector in feed direction, $\overrightarrow{b}$ is the unit vector in direction of tool pass interval.

2.2.3 Inverse kinematics transformation independent of machine configuration

Through the above calculations, CL data for five-axis machining has been obtained, which consists of a set of tool center position vector P and tool axis orientation vector O. In order to control the motion of machine tool, CL data in the coordinate system of workpiece must be transformed to the coordinates of every axis in the machine coordinate system through inverse kinematics transformation, which is dependent on specific configuration of machine tool. Here, the tilt–rotary table type five-axis milling machine is discussed. The two rotational coordinates axes refer to A and C, which are realized by worktable. Transformation equation is following:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(C) & -\sin(C) & 0 & wp_x + m_x \\ \cos(A)\sin(C) & \cos(A)\cos(C) & -\sin(A) & wp_y + m_y \\ \sin(A)\sin(C) & \sin(A)\cos(C) & \cos(C) & wp_z + m_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \tag{10}$$

$$\begin{aligned} A &= -\cos^{-1}\left(\frac{\sqrt{O_x^2 + O_y^2}}{\sqrt{O_x^2 + O_y^2 + O_z^2}}\right) \\ C &= \tan^{-1}\left(\frac{O_x}{O_y}\right) \end{aligned} \tag{11}$$

The workpiece offset vector, ***wp***, is the position of the workpiece coordinate system relative to the tilt-rotary table coordinate systems. The machine offset vector, ***m***, is the position of the workpiece coordinate system to the machine coordinate system. The parameters (X, Y, Z, A, C) are instruction values for the stages of a five-axis milling machine.

3 Development of a software STEP-NC controller

3.1 State of the art on STEP-NC controllers

Since the establishment of STEP-NC model of data, lots of research works have been carried out on the design and implementation of STEP-NC controller [16]. According to reference [19], the controller is classified into three types.

In the first type of STEP-NC controller, an interpreter of STEP-NC program is developed, and conventional controller can be used without modification. The first prototype was developed by European STEP-NC project, which designed a converter from STEP-NC files to G&M instructions added to the front-end of SIEMENS 840D controller. STEP Tools developed software tools as plug-ins to implement STEP-NC machining on industrial CNC machine tools. The National Institute of Standards and Technology (NIST) in the USA built two STEP-NC interpreters for milling operations: one using ISO 14649, the other using

AP 238 of ISO 10303, where STEP-NC program was interpreted to generate specific machine instructions [20]. Similar solution can be found in reference [21].

The second type of the STEP-NC controller can provide the function of tool path generation and realize machining based on manufacturing features. At POSTECH, the team led by Professor Suh developed an open platform called Korea STEP-NC, which included floor shop programming system, tool path generator, tool path viewer, man–machine–interface module and CNC kernel [22]. They had made efforts on turning application based on STEP-NC [23, 24]. At the University of Auckland, Xu et al. developed a STEP-NC controller with hierarchical architecture using function block [25]. A prototype system for turning application was also developed, which consists of three levels: input level, NC level, and signal level [26]. At IRCCyN in France, the platform, called SPAIM, which controls current industrial CNC machine tools directly from STEP-NC file was presented [27]. It includes HMI, tool paths generation module, CAD reconstruction module, simulation module, programming module, etc.

The third type of the STEP-NC controller is expected to be the intelligent, autonomous CNC controller. The ideas and methods on real-time monitoring and diagnosis, self-learning decision, adaptive optimization, close-loop machining based on STEP-NC have been proposed and experienced in the laboratories [28–30]. In the Foundation for the Factory of the Future (FoFdation) project funded by European Commission, a generalized, adaptive Smart Manufacturing Controller architecture is proposed, which satisfies both commercial and open source CNC controller. The use cases demonstrate optimization by simulation and in-situ quality and process monitoring. The FoFdation project is ongoing and its ultimate goal is higher productivity and quality with lower impact to the environment [31].

STEP-NC data model provides an effective data exchange mechanism between design systems and manufacturing systems. However, the progress on developing the true intelligent, autonomous STEP-NC controller is still slow, and more efforts need to be made.

3.2 Development platform and system structure

To develop a five-axis software CNC system for machining surfaces is the goal of the paper, i.e., the manner of software realization is applied. In the development, system platform and interface used are showed in Table 1 to follow the openness rule.

The structure of the STEP-NC controller is shown in Fig. 4. First, windows and RTX are adopted as software development platform. The main reason is that they fulfill the two requirements for openness and real-time.

STEP-NC is chosen as NC programming interface, which ensures geometric information and technologic information to be input to CNC entirely. Based on previous discussion, CNC system would become more intelligent. Naturally, much more tasks need to be developed.

In addition to the adoption of STEP-NC as NC programming interface, SERCOS protocol as an international standard interface also be used to exchange data between the CNC system and servo system, I/O devices. In the project, SERCOS communication is completed through SoftSERCANS card that is installed in PC. SoftSERCANS card is the product from Indramat. SoftSERCANS provides API in the manner of Dynamically Linked Library (DLL) to realize SERCOS communication. Using the API, program for data communication can be complied easily, for example, cyclic transmission of command values and feedback values. This manner is fully accord with the style of the project, software realization, which increases openness of CNC.

Based on above platform and standard communication interface, a five-axis CNC has been developed, which consists of a set of software modules. Both motion control and PLC control functions are included in the software CNC. The software modules are classified into real-time modules and non-real-time modules. HMI and STEP-NC Interpreter are COM components and run under Windows environment, Task Coordinator module, Axis Group module, and SoftPLC module are DLLs and run under RTX environment. The information exchange between real-time task and non-real-time task can be done by using shared memory. The function of main modules follows:

STEP-NC Interpreter interprets a part program into a series of structure data which include geometric parameters, coordinates, machining strategies, tool data, machine function, etc.

Table 1 System platform and interface of STEP-NC controller

Platform		API	Communication interface	
Hardware	Operating system		NC programming interface	Servo system and I/O
PC+communication card (Indramat SoftSERCANS card)	Windows+RTX (Real-Time eXtension for Windows)	Referencing OMAC partly [34]	STEP-NC	SERCOS

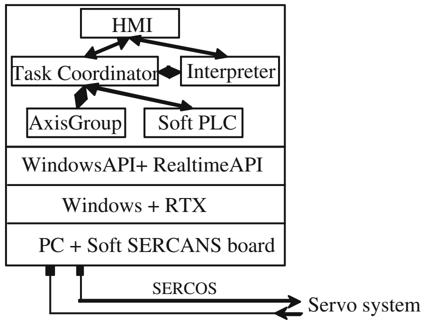


Fig. 4 The structure of the STEP-NC controller

Task Coordinator module gets interpretations through shared memory and finishes cutting path planning and planning for machine device turning on/off; Execution Step unit of two types can be obtained by planning, one of which is related to interpolation, another of which is related to discrete logic control. Task Coordinator in the system mainly serves as the center of management, which is responsible for coordination of all modules, switch of operation modes, and loading, startup, monitoring, shutdown, and unloading of real-time modules, etc.

Axis Group Module is responsible for real-time interpolation.

SoftPLC module is responsible for reading and writing status of I/O devices periodically, and logic calculation.

In addition, SERCOS communication object based upon SoftSERCANS technology is a component placed in Task Coordinator, which provides a SERCOS standard interface to exchange data between CNC, servo system, and I/O devices.

3.3 Module collaboration and design of data flow

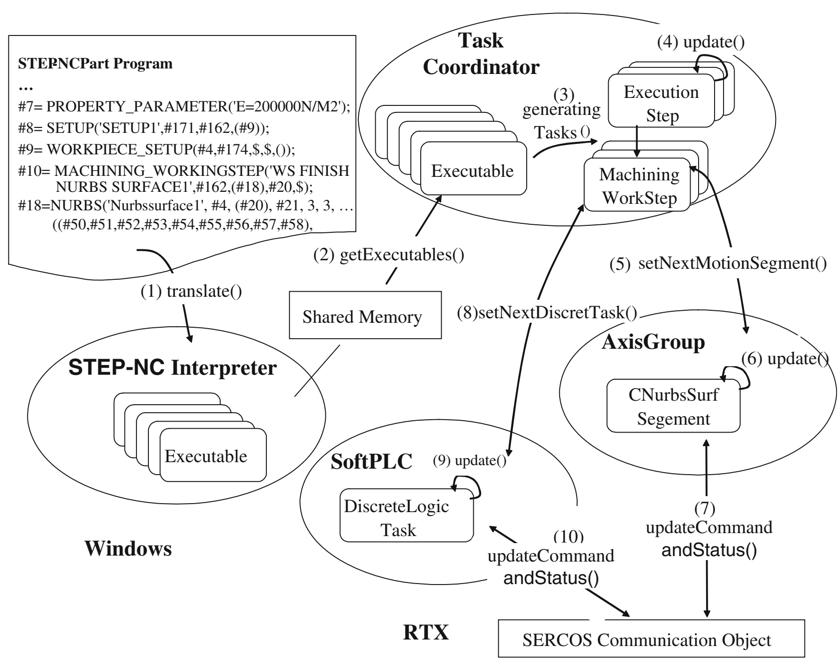
Modularity is the key features in the system. The coordination of modules, interoperation among modules and the design of data flow are all the major issues need to be solved in the development of system [32]. Figure 5 is used to explain in the STEP-NC controller how one STEP-NC program for NURBS surface machining is dealt with, modules collaborate, and data flow is transferred in auto mode.

In step 1, the Interpreter module translates a STEP-NC part program into a series of structure data—executables including all of geometric and technological information, where the contents correspond to the definition of entity working step and entity nc_function in ISO 14649 [33].

In step 2, Task Coordinator uses getExecutables() to retrieve all of interpretation results. In the system, the Task Coordinator module running in RTX and the Interpreter executing in Windows exchange data using share memory, and keep synchronization using interprocess communication object such as mutex, semaphore, etc.

In step 3, after Task Coordinator deals with the interpretation by calling the method generatingTask() to complete cutting path planning and discrete logic task planning, executables are converted into a series of Execution Step Units which are of nested finite state machine data and can be classified different types according to the difference of its content. Execution Step Unit is of embedded finite state machine, which mainly contains task units of two types, MachiningworkStep unit (for example, CNurbsSurfSegement which is embedded in CNurbsSurfExecuteStep) related to

Fig. 5 Collaboration of modules in CNC



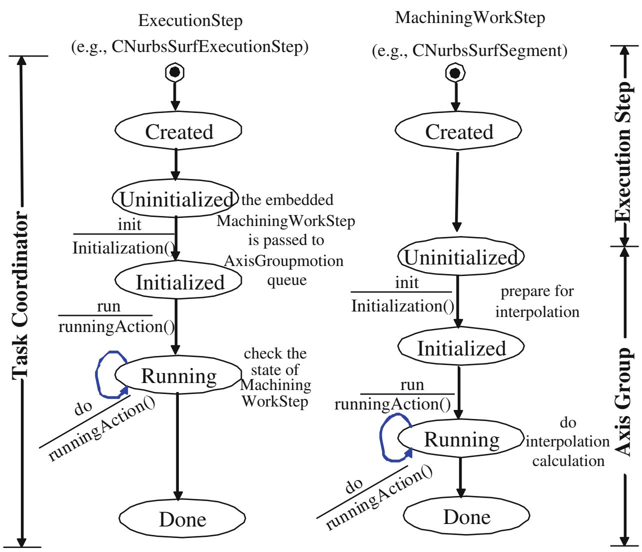


Fig. 6 Execution procedure of the nested task units in CNC

interpolation and DiscreteLogicTask unit (for example, CCoolantTurnOn) related to discrete logical control.

In step 4, Task Coordinator triggers execution of Execute Step Units' finite state machine. Step 4 may be repeated several times as in the case where the Execution Step may have to synchronize with lower level modules (e.g., such as waiting until the current NURBS surface interpolation have first completed).

In step 5, Execution Step unit in active state appends the embedded machiningWorkStep to motion queue in Axis Group module using the method setNextMotionSegment(). Once the MachiningWorkStep unit is loaded onto the Aixs Group queue, it waits for activation.

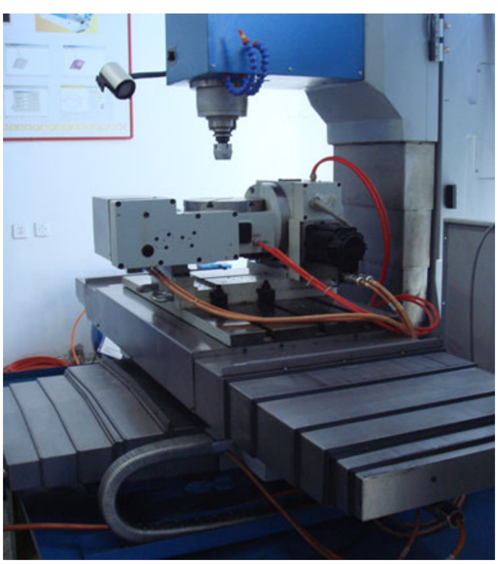


Fig. 7 Five-axis machine tool

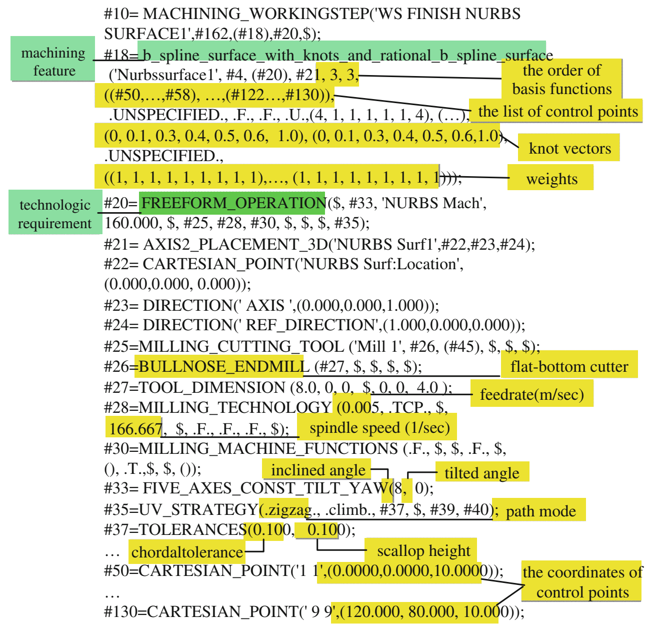


Fig. 8 The part program for five-axis machining of a NURBS surface

In step 6, once activation, machiningWorkStep's update() method runs periodically in every interpolation cycle. At the same time, the finite state machine of outer level Execution Step unit is running in Task Coordinator to monitor the execution of the machiningWorkStep, until interpolation of the machiningWorkStep is completed, converted to Done state. The Next Execution Step enters into active state, repeating steps 4 or 7.

In step 7, Axis Group sends position commands to SERCOS Communication Object which is responsible of bi-direction data exchange between CNC and servo system by calling the method updateCommandAndStatus().

From step 8 to step 10, the work similar to steps 4 to 7 is done. In Task Coordinator, DiscreteLogicTasks which are embedded in Execution Step in order are passed to SoftPLC module. Once DiscreteLogicTask active, the update() method runs automatically in every PLC cycle. At the same time, its outer level Execution Step also executes to poll the DiscreteLogicTask's state, until DiscreteLogicTask converts to Done state. SoftPLC update output devices' status by SERCOS communication object. Similarly, the Next Execution Step enters into active state, repeating steps 4 or 8.

Figure 6 shows the state transition of MachiningWorkStep and the change in management object of MachiningWorkStep when it is transferred and executed. First, nested Execution Step units are created in Task Coordinator. Task Coordinator activate finite state machine of outer level Execution Step, using Execution Step's method update() to pass the embedded MachiningWorkStep to Axis Group motion queue and to make the MachiningWorkStep entering to running state. In the

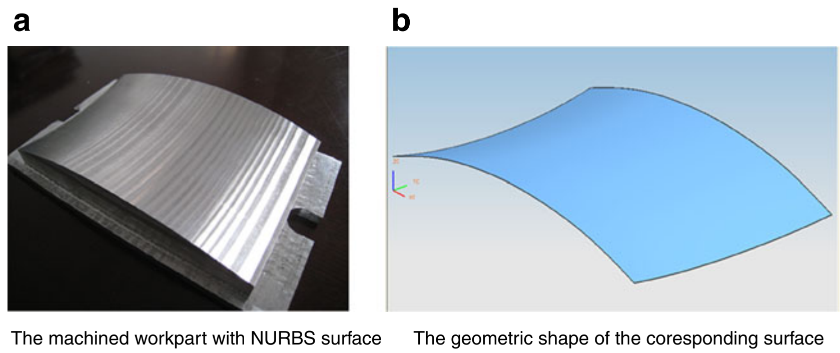


Fig. 9 The machined work part; **a** the machined work part with NURBS surface, **b** the geometric shape of the corresponding surface

above process, embedded MachiningWorkStep state doesn't change. Then, Axis Group module calls the MachiningWorkStep's method update(), where the MachiningWorkStep finite state machine handles with the event in it and the event run orderly to activate interpolation calculation until MachiningWorkStep enters into the state Done. At the same time, outer level Execution Step in Task Coordinator still keeps in state running, and periodically polls whether the embedded MachiningWorkStep has been completed. Once completed, Axis Group deletes the MachiningWorkStep from motion queue; at the same time, Task Coordinator makes the next Execution Step into the active state. On the two sides in Fig. 6, the corresponding management objects of Execution Step and MachiningWorkStep are showed at different time.

From the above example, it shows that, using the design of Execution Step with nested structure, Task Coordinator not only ensure NC data is exchanged orderly and accurately, but also coordinate and monitor the lower level modules to realize the expected functions.

4 The five-axis milling experiment

The five-axis software STEP-NC controller has been developed and executed on a five-axis machine tool with tilt–rotary table, as shown in Fig. 7.

In the machine tool, the milling of one workpart with NURBS surface has been performed. The surface has the shape of saddle, and is modeled by 9×9 control points. The milling experiment was carried out under the following condition: spindle speed, 1,000 r/min; feed rate, 300 mm/min; flat–bottom cutter with radius, 4 mm; inclined angle, 8°; tilted angle, 0°; scallop height, 0.1 mm. These geometric information and technological information are entirely included in a part program written in STEP-NC. A part of the part program is shown in Fig. 8. The machined work part and the shape of the corresponding NURBS surface are shown in Fig. 9.

The following conclusions have been drawn through the experiment.

The structure of STEP-NC controller is well designed, the expected functions have been completed, the coordination among modules is good, the data exchange is accurate.

The proposed method for five-axis machining of NURBS surface based on STEP-NC proved accurate and valid, which overcomes the limitations in conventional method.

The STEP-NC controller has capability of cutting path planning and generation, and meeting requirement on real-time. Currently, the interpolation cycle time is set 2 ms. In the milling experiment, the real longest time of interpolation calculation is 0.138 ms (approximately accounting for 6.9 %).

5 Conclusions

The open system platform and the standard interfaces have been adopted to develop the STEP-NC controller, which improve the openness of controller. To realize of the machining freeform surfaces, the description of NURBS surface in STEP-NC standard is investigated. A five-axis STEP-NC prototype system with a series of software has been built, where the functions of path planning, CC calculation, tool offset and inverse kinematics transformation are transferred from CAM to controller. Milling experiment has been performed. Experimental results prove the proposed method valid.

The work provides an effective, new method in the machining freeform surfaces, and demonstrates the potential of STEP-NC data model on the machining of freeform surfaces.

Acknowledgments The authors gratefully acknowledge the supports from National Natural Science Foundation of China (no. 50675002), National Natural Science Foundation of China (no. 51075078) and the Scientific Research Foundation for the Returned Overseas Chinese Scholars of State Education Ministry.

References

1. Ochiai Y, Sekiya T (1995) Generation of free-form surface in CAD for dies. Adv Eng Softw 22(2):113–118
2. Huang Y, Oliver JH (1994) Non-constant parameter NC tool path generation on sculptured surface. Int J Adv Manuf Technol 9(5):281–290
3. Yeh SS, Hsu PL (2002) Adaptive-feedrate interpolation for parametric curves with a confined chord error. Comput Aided Des 34:229–237
4. Tutunea-fatan OR, Feng HY (2005) Determination of geometry-based errors for interpolated tool paths in five-axis surface machining. J Manuf Sc and Eng 127(1):60–67
5. Tsai MS, Nien HW, Yau HT (2008) Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery. Comput Aided Des 40(5):554–566
6. Xu ZM, Chen JC, Feng ZJ (2002) Performance evaluation of a real-time interpolation algorithm for NURBS curves. Int J Adv Manuf Technol 20(4):270–276
7. You YP, Wang M (2001) An interpolator for NURBS curve machining with high speed and high accuracy. J Comput-aided Des Comput Graph 13(10):943–947
8. Zhang QG, Greenway RB (1998) Development and implementation of a NURBS curve motion interpolator. Robot Comput-Integr Manuf 14(1):27–36
9. Cheng CW, Tsai MC (2004) Real-time variable feed rate NURBS curve interpolator for CNC machining. Int J Adv Manuf Technol 23(11–12):865–873
10. Park J, Nam S, Yang MY (2005) Development of a real-time trajectory generator for NURBS interpolation based on the two-stage interpolation method. Int J Adv Manuf Technol 26(4):359–365
11. Liang HB, Wang YZ, Li X (2006) Research and implementation of NURBS interpolation algorithm for adaptive feed speed. Compt Integr Manuf Sys 12(3):428–432 (In Chinese)
12. Cheng CW, Tseng WP (2006) Design and implementation of a real-time NURBS surface interpolator. Int J Adv Manuf Technol 30(8):98–104
13. Spence A, Altintas Y (1991) CAD assisted adaptive control for milling. Trans ASME J Dyn Syst Meas Control 113(3):444–450
14. Jerard RB, Ryou O (2006) NCML: a data exchange format for internet-based machining. Int J Comp App in Tech 26(1/2):75–82
15. ESPRIT Project EP29708. (2001) STEP-NC final report of Europe STEP-NC.
16. Xu XW, Wang H, Mao J, Newman ST, Kramer TR, Proctor FM (2005) STEP-compliant NC research: the search for intelligent CAD/CAPP/CAM/CNC integration. Int J Pro Res 43(17):3703–3743
17. Suh SH, Cheon SU (2002) A framework for an intelligent CNC and data model. Int J Adv Manuf Technol 19(10):727–735
18. STEP Manufacturing Forum. http://www.steptools.com/manforum/meetings.html. Accessed July 2012
19. Suh SH, Lee BE, Chung DH, Cheon SU (2003) Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. Comput Aided Des 35:1069–1083
20. Kramer TR, Proctor FM, Xu X, Michaloski JL (2006) Run-time interpretation of STEP-NC: implementation and performance. Int J Comput Integr Manuf 19(6):495–507
21. Xu XW (2006) Realization of STEP-NC enabled machining. Robot Comput-Integr Manuf 12(2):144–153
22. Suh SH, Chung DH, Lee BE, Cho JH, Cheon SU, Hong HD, Lee HS (2002) Developing an integrated STEP-compliant CNC prototype. J Manuf Sys 21(5):350–362
23. Suh SH, Chung DH, Lee BE, Shin SS, Choi I, Kim KM (2006) STEP-compliant CNC system for turning: data model, architecture, and implementation. Comput Aided Des 38(6):677–688
24. Chung D, Suh SH (2008) ISO 14649-based on nonlinear process planning implementation for complex machine. Comput Aided Des 40:521–536
25. Minhat M, Vyatkin V, Xu X (2009) A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks. Robot Comput-Integr Manuf 25:560–569
26. Habeeb S, Xu X (2009) A novel CNC system for turning operations based on a high-level data model. Int J Adv Manuf Technol 43:323–336
27. Rauch M, Laguionie R, Hascoët J-Y, Suh S-H (2012) An advanced STEP-NC controller for intelligent machining processes. Robot Comput-Integr Manuf 28:375–384. doi:10.1016/j.rcim.2011.11.001
28. Brecher C, Vitr M, Wolf J (2006) Closed-loop CAPP/CAM/CNC process chain based on STEP and STEP-NC inspection tasks. Int J Comput Integr Manuf 19(6):570–580
29. Valente A, Carpanzano E, Nassehi A, Newman ST (2010) A STEP compliant knowledge based schema to support shop-floor adaptive automation in dynamic manufacturing environments. CIRP Ann Manuf Technol 59:441–444
30. Ridwan F, Xu X, Liu G (2012) A framework for machining optimisation based on STEP-NC. J Intell Manuf 23:423–441. doi:10.1007/s10845-010-0380-9
31. Hentz JB et al (2012) An enabling digital foundation towards smart machining. 8th CIRP Conf Intell Comput Manuf Eng. doi:10.3929/ethz-a-007590615
32. Li X. (2005) Research and development of an open Soft-CNC system and related technology. Dissertation of PhD, Harbin Institute of Technology, China. (In Chinese)
33. International Standards Organization (2002) ISO 14649–10. Data model for computerized numerical controllers: part 10—general process data
34. OMAC Working Group. OMAC API Reference Documentation