Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 9814973, 14 pages
http://dx.doi.org/10.1155/2016/9814973

Research Article

Implementation of Real-Time Machining Process Control Based on Fuzzy Logic in a New STEP-NC Compatible System

Po Hu, Zhenyu Han, Yunzhong Fu, and Hongya Fu

School of Mechatronics Engineering, Harbin Institute of Technology, 92 West Dazhi Street, Nan Gang District, Harbin 150001, China

Correspondence should be addressed to Hongya Fu; hongyafu@hit.edu.cn

Received 15 March 2016; Revised 18 April 2016; Accepted 5 May 2016

Academic Editor: Chuanxi Qian

Implementing real-time machining process control at shop floor has great significance on raising the efficiency and quality of product manufacturing. A framework and implementation methods of real-time machining process control based on STEP-NC are presented in this paper. Data model compatible with ISO 14649 standard is built to transfer high-level real-time machining process control information between CAPP systems and CNC systems, in which EXPRESS language is used to define new STEP-NC entities. Methods for implementing real-time machining process control at shop floor are studied and realized on an open STEP-NC controller, which is developed using object-oriented, multithread, and shared memory technologies conjunctively. Cutting force at specific direction of machining feature in side mill is chosen to be controlled object, and a fuzzy control algorithm with self-adjusting factor is designed and embedded in the software CNC kernel of STEP-NC controller. Experiments are carried out to verify the proposed framework, STEP-NC data model, and implementation methods for real-time machining process control. The results of experiments prove that real-time machining process control tasks can be interpreted and executed correctly by the STEP-NC controller at shop floor, in which actual cutting force is kept around ideal value, whether axial cutting depth changes suddenly or continuously.

1. Introduction

Machining efficiency and quality of finished parts can be improved by monitoring, analyzing, and diagnosing the process of product manufacturing. It is hard to model the machining process accurately due to its complexity and variability. Therefore, artificial intelligent algorithms are usually used to build the relational model of machining parameters, cutting tool wear, and quality of finished part, with the purpose of developing machining process controllers that shorten machining time, prevent damage of tools, and improve quality of finished part. Li et al. used back propagation neural network for multiobjective cutting parameters optimization in sculpture parts machining to increase surface quality [1]. Huang et al. proposed a fuzzy control strategy based on constraint of spindle power in end milling process for reducing machining time of complex shape machining [2]. Zuperl et al. proposed an adaptive control strategy based on neural network to maximize the feed rate subject to allowable cutting force on the milling tool [3]. Off-line optimization and on-line adaptive adjustment based on neural control scheme (NCS) are combined to control the cutting force [4]. Research works have also been carried out to analyze and improve the performance of artificial intelligence algorithms. Osaba et al. presented a Standstill & Parade strategy for subpopulations communication in parallel genetic algorithms [5]. Precup et al. introduced an approach for analyzing the stability of nonlinear processes controlled by Takagi-Sugeno fuzzy logic controllers [6]. Qian proposed and studied the global attractivity of periodic solutions for nonlinear difference equation [7]. Guerra and Vermeiren analyzed the stabilization of nonlinear systems that can be modeled by Takagi and Sugeno (TS) discrete fuzzy models by using nonquadratic Lyapunov functions in [8]. From above publications, it could be concluded that the result of optimization or adjustment is greatly affected by the rapidity of machining process control, which is expected to be implemented as soon as possible, which means in real-time at shop floor in this case. However, two bottlenecks limit the rapidity of machining process control. One is to transfer high-level product manufacturing data between CAPP systems and CNC systems. The other is to embed real-time adaptive control algorithms or methods in the kernel of implementation platform.

The standard of STEP-NC supports bidirectional data transmission between CAD/CAM systems and CNC systems, which provides a possible way for solving the first bottleneck. Kumar et al. presented a STEP-NC compliant process control framework for discrete components and relevant self-learning algorithms in order to compensate errors and improve the surface quality of finished part [9]. Laguionie et al. proposed a STEP-NC-compliant manufacturing scenario to optimize the process routes in multiprocess manufacturing environment [10]. Campos and Hardwick proposed a feature-based traceability approach based on STEP-NC in order to feed back the manufacturing data associated with the monitored processes [11]. Then the traceability interface for STEP-NC is defined and associated with other standards such as ISA-95 and MTConnect to support traceability activities in collaborative manufacturing scenario [12]. Wosnik et al. presented a STEP-NC-compliant data model and process chain architecture for the optimization of machining process based on feedback process data [13]. Kumar et al. established the information models and implementation mechanism for machine tool process control based on STEP-NC [14]. Ridwan et al. proposed the STEP-NC data models and relevant optimization algorithm for real-time process control and monitoring, so that the high-level machine condition monitoring can be used for optimizing machining process [15]. Machine condition monitoring (MCM) based on STEP-NC standard is proposed, which enables optimization during machining in order to shorten machining time and increase product quality. The MCM system consists of three modules, namely, optiSTEP-NC, AECopt, and knowledge-based evaluation (KBE) [16]. However, most of the machining process control methods based on STEP-NC, implemented at machining process stage, are still off-line. It is difficult for a few on-line adaptive control algorithms, which are implemented on external machining process controllers at shop floor, to realize real-time machining process control. The root of this problem is the above publications did not find suitable platform for solving the second bottleneck.

In this paper, a framework and implementation methods of real-time machining process control based on STEP-NC are studied. STEP-NC data model for real-time machining process control is defined in order to transfer high-level product information between CAD/CAM systems and CNC systems. A software-based STEP-NC controller, which interprets and executes high-level information in STEP-NC files, such as manufacturing features, machining operations, and real-time machining process control functions, is designed and developed. An adaptive control algorithm based on fuzzy logic is also embedded within the software kernel of STEP-NC controller in order to improve the real-time performance of machining process control. The STEP-NC data model, software STEP-NC controller, and implementation methods proposed in this paper can be used to realize real-time machining process control at shop floor. The rest of the paper is organized as follows. In Section 2, the architecture of different types of machining process controllers is analyzed and compared. In Section 3, the STEP-NC data model and implementation methods for real-time machining process control are proposed, along with the design of STEP-NC controller that is able to interpret the newly defined entities. In Section 4, the fuzzy control algorithm for real-time machining process control is introduced and validated. Finally, experiments are carried out to verify the proposed framework, STEP-NC data model, and implementation methods for real-time machining process control.

2. Architecture of Machining Process Controller

Machining process control methods are usually designed based on automatic control theories such as classical control theory, modern control theory, information theory, system theory, and artificial intelligence theory. Machining process control systems become more integrated, interoperable, and intelligent with the development of computer and network technologies in recent years. The procedure of machining process control consists of three stages, namely, information collection, information processing, and control output. According to real-time property, machining process control can be classified into three categories.

(1) Off-Line Machining Process Control. Machining process information is acquired and saved during or at a certain stage of machining process. Then the acquired data is saved and analyzed by external machining process controllers, which adjust machining parameters of later machining process plans.

(2) On-Line Machining Process Control. Machining process information is acquired in real-time or at a certain stage of machining process. Then the machining process is paused to analyze the data and evaluate the process condition. Cutting parameters of subsequent working steps are adjusted to fix or optimize the machining process.

(3) Real-Time Machining Process Control. Machining system executes tasks of motion control, position control, data acquisition, data analyzing, and parameter adjusting in every interpolation period in order to keep the machining condition at desired and ideal state, which means all tasks of part machining, machining condition monitoring, and adaptive control are executed in real-time.

Machining process controllers (MPC), which analyze the process condition and adjust machining parameters, can be implemented at process planning stage or shop floor stage as shown in Figure 1. For MPC at process planning stage, CNC systems execute motion control command without adjustment even if the machining process is unstable. Optimization can only be done when the next part is being machined. For external MPC at shop floor, machining process data is firstly used for adjusting machining parameters on-line and then transferred to CAPP systems for further analysis in order to optimize process plan. MPC at shop floor is usually

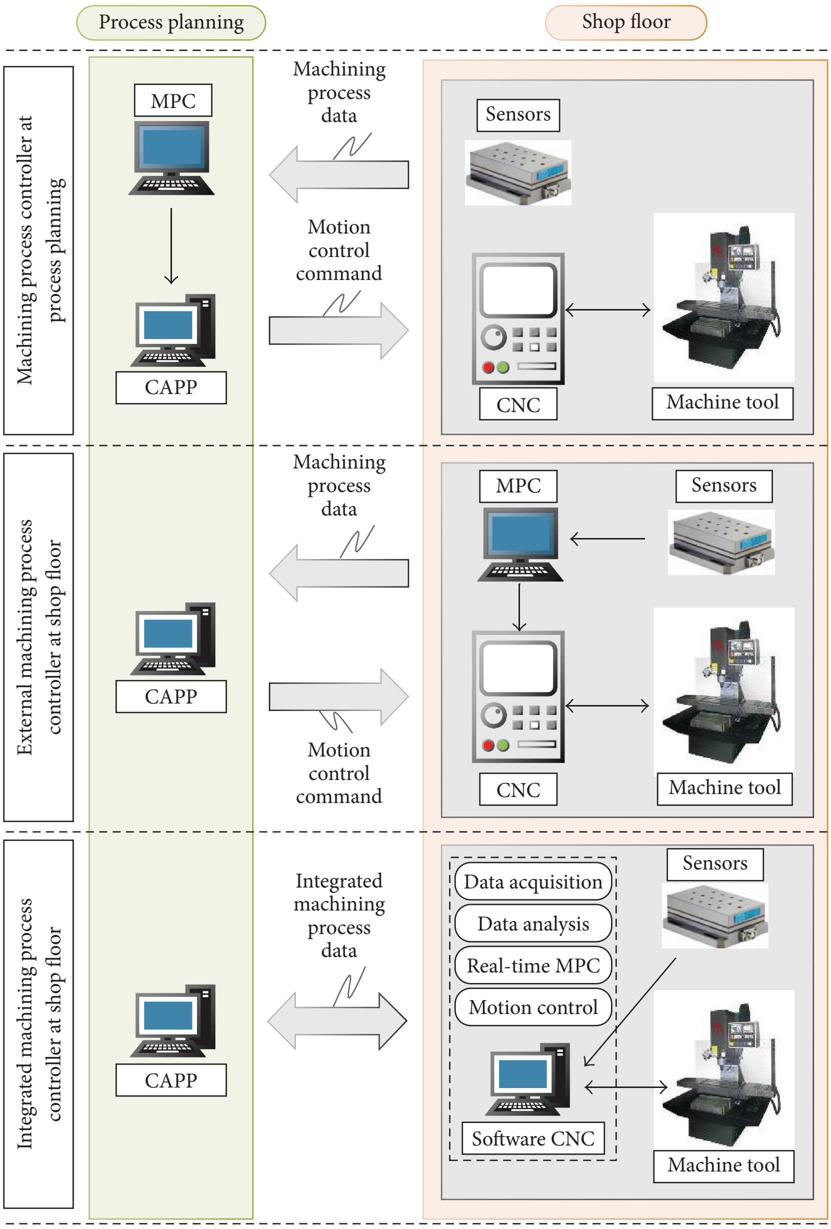


Figure 1: Implementation of machining process controller.

implemented on an external computer, which sends adjusted machining parameters to CNC systems. For integrated MPC at shop floor, CNC systems execute machining tasks by interpreting high-level integrated machining process data directly and include dimension and position of manufacturing features, machining methods, and process parameters. Machining process data is acquired and analyzed by real-time machining process control algorithms simultaneously with interpolation computing. The data acquired during machining process is integrated with input data and sent to CAPP system in order to preserve manufacturing knowledge for later machining process planning. Separation of MPC and CNC systems will lead to the delay of machining process control, which means integrated MPC at shop floor is a possible solution for implementing real-time machining process control.

3. Real-Time Machining Process Control Based on STEP-NC

3.1. Problem Definition and Analysis. Fluctuation of cutting force has great influence on machining system stability, cutting tool life, dimensional accuracy, and surface quality of finished part. It is necessary to select proper machining parameters with constraint of maximum or optimal cutting force. In this paper, the control of cutting force at specific direction is chosen as controlled object to implement real-time machining process control. As shown in Figure 2, side

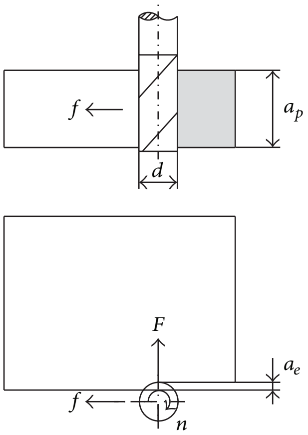


Figure 2: Side milling with end mill.

milling of planar face with end milling tool is taken as an example. The cutting force F normal to the machined surface, which will cause the deformation of cutting tool, workpiece, and fixture, is selected as the controlled object. Optimal cutting force is calculated by CAD/CAM system and sent to CNC system at shop floor along with other process planning data. The prediction algorithm of differential tangential (dF_t), radial (dF_r), and axial (dF_a) cutting forces is given by Engin and Altintas in [17] as

$$\begin{aligned} dF_t &= K_{te}dS + K_{tc}h(\phi,\kappa)\,db, \\ dF_r &= K_{re}dS + K_{rc}h(\phi,\kappa)\,db, \\ dF_a &= K_{ae}dS + K_{ac}h(\phi,\kappa)\,db. \end{aligned} \tag{1}$$

Edge cutting coefficients K_{te}, K_{re}, and K_{ae} and shear force coefficients K_{tc}, K_{rc}, and K_{ac} are various in different cutting conditions, which can only be identified from plenty of cutting tests. It is hard to build an accurate model for cutting process as the interrelation of machining parameters and cutting force is too complicated. As a result, the control algorithms based on empirical equation and off-line optimization usually become inappropriate at real cutting process, because of tool wear, varying cutting conditions, and complex shape of part. In this paper, real-time optimization algorithm is used to calculate the ideal machining parameters for current cutting condition by analyzing actual cutting force signal, which is acquired by a dynamometer. There are four machining parameters that are mainly related to cutting force, namely, axial cutting depth (a_p), radial cutting depth (a_e), feed rate (f), and spindle speed (n). Axial cutting depth a_p and radial cutting depth a_e are determined by machining allowance and shape of part, which are usually hard to be adjusted in real-time during the cutting process. Feed rate f, which has more influence on cutting force than spindle speed n, is adjusted in real-time in order to keep actual cutting force equal to optimal value. Acquired cutting force data should also be preserved and sent to machining knowledge management system for further analysis. Therefore, a data model for describing and transferring information of real-time machining process control tasks and machining process condition parameters is needed.

3.2. STEP-NC Data Model. STEP-NC data transfer standard describes the machining process plan with entities such as *project*, *workplan*, *workpiece*, *machining_workingstep*, *machining_feature*, *machining_operation*, *machining_function*, and *technology*, which are defined by using EXPRESS language, a basic descriptive method of STEP-NC standards [18]. Machining system gathers information of machining condition and associates it with entities of machining process plan. However, there is no data model for real-time machining process control and machining process condition information in STEP-NC data transfer standard at present. To pursue the aim of this research, which controls the cutting force of side mill perpendicular to machined surface by adjusting feed rate in real-time, new STEP-NC entities for describing constant force milling function and cutting force data are defined by using EXPRESS method in this paper. Figure 3 is the EXPRESS-G diagram of the developed data model for real-time machining process control. The upper half gives a brief description of existing STEP-NC data model, which is referenced from relevant ISO 14649 Part 10, Part 11, and Part 111 [19–21]. The lower half expresses the newly defined STEP-NC data model, which is compatible with existing STEP-NC standards of ISO 14649.

Entity *const_force_milling* is a subtype of *adaptive_control*, which is defined in ISO 14649 Part 11. Under this entity there are six attributes, *x_axis_const*, *y_axis_const*, *z_axis_const*, *x_force*, *y_force,* and *z_force*. The first three attributes represent the direction of milling force that is supposed to be constant. Then the last three attributes represent the ideal milling force in Newtons. The direction and magnitude of milling force, which is applied to machined surface by milling cutting tool, are defined as three components at x, y, or z direction in feature coordinate system. The attributes of milling force are optional if the value of corresponding Boolean attribute is *False*. When the *adaptive_control* attribute of entity *machining_technology* in STEP-NC file points to an instance of *const_force_milling*, the STEP-NC controller will execute the corresponding working step with variable machining parameters in order to keep the milling force at optimal value. Meanwhile, real-time intelligent control algorithm is used to realize machining process control.

Entity *milling_force_save* is defined to preserve the cutting force data. There are six attributes, *its_workingstep*, *start_point*, *end_point*, *sample_rate*, *local_save,* and *online_save*. The cutting force data acquired during cutting process is used for optimizing machining parameters in real-time. Then the acquired data should be associated with corresponding working step and sent to other subsystems for further analysis. Attribute *its_workingstep* represents the corresponding *machining_workingstep* that is being executed while the cutting force data is being acquired. Attributes *start_point* and *end_point* represent the positions of milling cutting tool in feature coordinate system at the beginning and the end of data acquisition. Attribute *sample_rate* represents the sampling period of cutting force acquisition in microseconds. Total

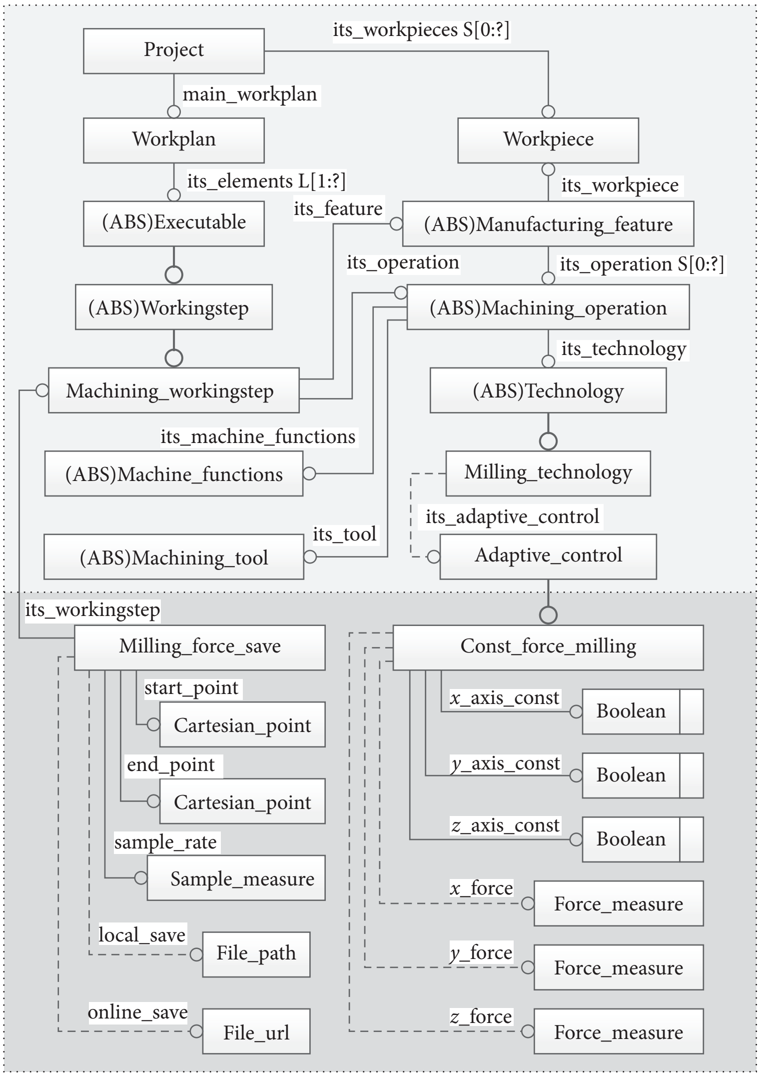


FIGURE 3: EXPRESS-G diagram of real-time machining process control.

integration of product manufacturing information can be realized if the real-time machining process condition data is saved in STEP-NC file. However, the quantity of cutting force data is usually too huge for STEP-NC file, especially when the sampling frequency is very high. In this case, the cutting force data is saved in an individual file that is stored in local storage of web server, while the file path or URL is represented as character strings by attribute *local_save* or *online_save*. At least one out of the last two optional attributes must exist.

STEP-NC file contains high-level information of product manufacturing without low-level tool path, in which entity *project* is the root node of tree structure. CNC systems should interpret all instances in an input file in order to get necessary information for on-line tool path generation and execution. The first step of interpreting a STEP-NC file is to map the information of input file to internal data model. C++ classes are defined according to the EXPRESS definition of newly developed STEP-NC entities and added to the ISO 14649 class library, which is originally designed by NIST for off-line STEP-NC interpreting [22]. It is modified and embedded to the software CNC kernel of STEP-NC controller developed in this paper to interpret STEP-NC files on-line. Figure 4 presents the C++ classes for STEP-NC real-time machining process control information. Class *iso14649CppBase* is the

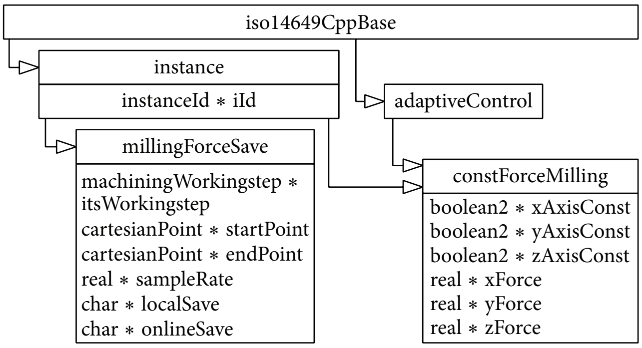


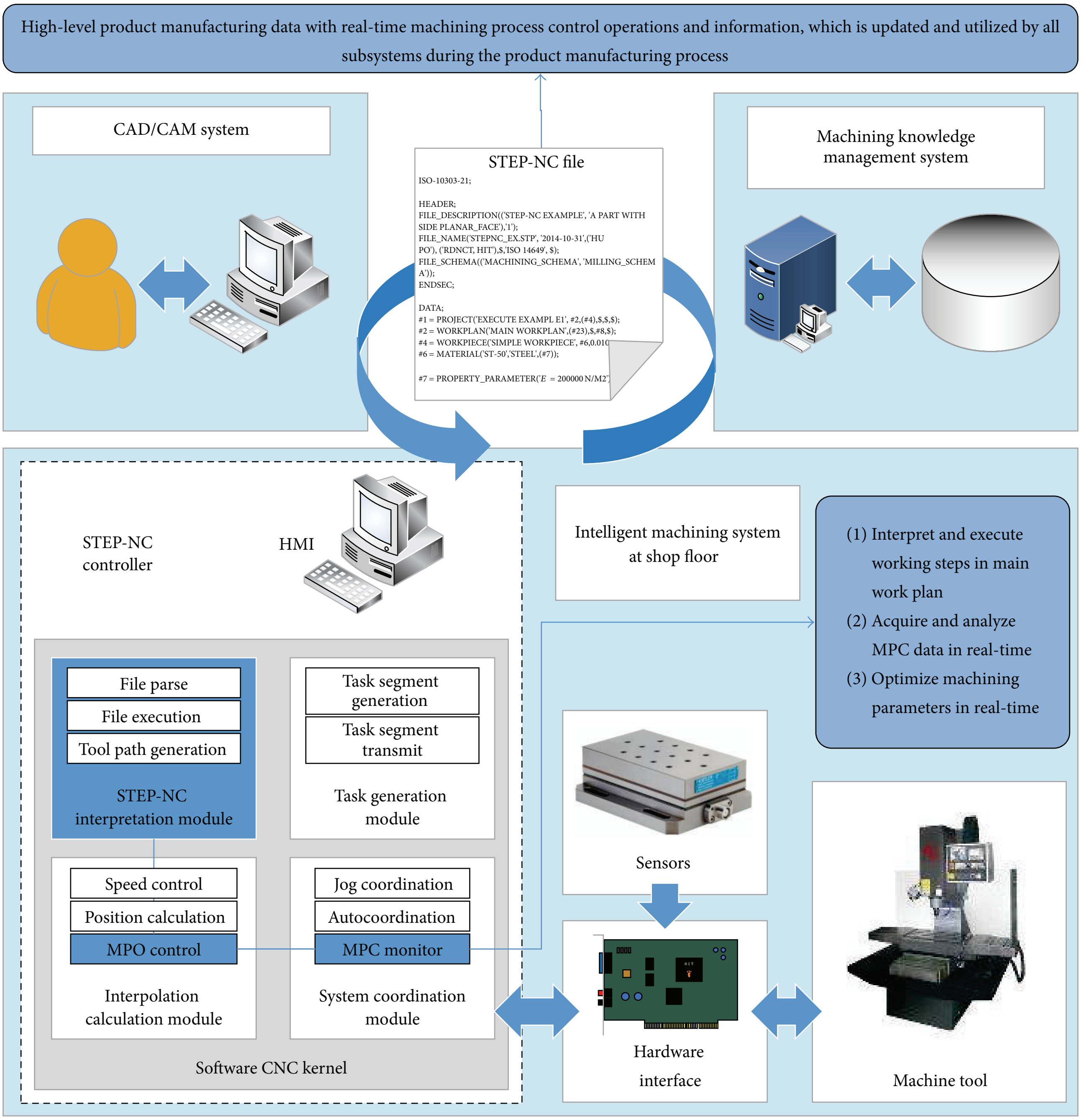
FIGURE 4: C++ classes for STEP-NC real-time machining process control information.

base class for all other classes in the class library. Class *instance* is the base class for C++ classes that maps STEP-NC instances in STEP-NC files. Member variable *iId* of *instance* represents the number of instances. Member variables of C++ classes are pointers to instances of other classes, which represent STEP-NC types or entities. By this way, nodes in tree structure of STEP-NC file are linked to each other.

3.3. Open STEP-NC Controller. CNC systems should be able to interpret the STEP-NC files that contain machining process control operations directly, acquire machining process condition data in real-time, and compute interpolation points and optimized machining parameters simultaneously. Most commercial CNC systems have limited interfaces for STEP-NC interpretation and real-time adaptive control. In this paper, A STEP-NC controller based on open architecture software CNC kernel is proposed and developed in order to implement real-time machining process control at shop floor. An integrated data model based on STEP-NC is used to describe the information of geometry, technology, process planning, and machining process condition, which makes all stages of product manufacturing process traceable. The STEP-NC controller interprets STEP-NC files directly while communicating with sensors without external data acquisition and analysis system. Adaptive control algorithms can be embedded into interpolation calculation procedure in order to optimize machining parameters. The architecture of machining process control system that consists of three subsystems is shown in Figure 5, in which ISO 14649 is used as data transfer standard for real-time machining process control. The first subsystem is CAD/CAM system, which is responsible for making the machining process plan that contains real-time machining process control functions. Then the machining process plan is sent to other subsystems in form of STEP-NC files. Machining knowledge got from former product manufacturing process, machine tool capability, and cutting tool condition are considered by CAD/CAM system to optimize the process routes and parameters off-line in order to get better quality of finished part or higher production efficiency. The second subsystem is machining knowledge management system, which is responsible for gathering, fusing, and managing product manufacturing data. The third subsystem is intelligent machining system, which is responsible for interpreting and executing STEP-NC file and optimizing machining process in real-time. At the core of this subsystem is an open STEP-NC controller developed by Research Division of Numeric Control Technology in Harbin Institute of Technology based on Open Modular Architecture Controller (OMAC). The function of STEP-NC controller is realized by a software CNC kernel that consists of four software modules, namely, STEP-NC interpretation module, task generation module, system coordination module, and interpolation calculation module. This paper focuses on the implementation of real-time machining process control at shop floor, and the realization of CAD/CAM system and machining knowledge management system goes beyond the scope of this study.

STEP-NC interpretation module is responsible for interpreting and executing STEP-NC files that contains real-time machining process control functions directly. Most of the manufacturing features derived from entity *machining_feature* in ISO 14649 Part 10, process data for milling in ISO 14649 Part 11, and milling cutting tool data in ISO 14649 Part 111 are supported by this module [19–21]. Tool paths are generated on-line by STEP-NC interpretation module and compiled by task generation module to generate task segments such as line feed, arc feed, and rapid move, which are transmitted to system coordination module via temporary storage. System coordination module creates several real-time threads for automatic operation, jog operation, and machining process condition monitoring, which are managed and synchronized by the host process according to scheduled clock period. System coordination module monitors the task segment queue in temporary storage in real-time, extracts one task segment each time when the task segment queue is not empty, and executes it by calling the functions of interpolation calculation module. Interpolation calculation module is responsible for optimizing machining process parameters and calculating the position of interpolation points, which consists of algorithms for speed control, position calculation, and MPC control. Under the interoperation of the four modules in software CNC kernel, the STEP-NC controller can parse and map a STEP-NC project into tree structure of C++ class instances, execute the main work plan, generate and execute tool paths for every working step, and control the machining process in real-time. Besides the software CNC kernel, a software Human Machine Interface (HMI) is developed for receiving instructions from user or communicating with other subsystems, and the hardware interface is used to communicate with servo system and sensors.

3.4. Procedure of Real-Time Machining Process Control Based on STEP-NC. Realization methods of real-time machining process condition monitor along with algorithms for real-time machining process control are studied and developed based on the open STEP-NC controller proposed and built in Section 3.3. A real-time thread is created by system coordination module to acquire machining process data from sensors via hardware interface, and the real-time adaptive control algorithm is embedded into interpolation calculation module to synchronize the procedure of machining parameters optimization and interpolation calculation. Procedure of real-time machining process control based on STEP-NC is shown in Figure 6. There are two host processes in the software CNC kernel of STEP-NC controller. The first process is responsible for interpreting STEP-NC file and generating tool paths. Tool paths are encapsulated as task segments, which contain information of real-time machining process control such as start point, end point, feed rate, spindle speed, and ideal cutting force. Task segments are pushed into a first in, first out (FIFO) task segment queue in shared memory. The second process is responsible for executing the task segment in real-time. When the task segment queue is not empty, which means the process of STEP-NC interpreting has started, the real-time process will get the first task segment from the queue and execute it until the queue is empty. Real-time machining process control algorithm is called to analyze



```
ISO-10303-21;

HEADER;
FILE_DESCRIPTION(('STEP-NC EXAMPLE', 'A PART WITH
SIDE PLANAR_FACE'),'1');
FILE_NAME('STEPNC_EX.STP', '2014-10-31',('HU
PO'), ('RDNCT, HIT'),$,'ISO 14649', $);
FILE_SCHEMA(('MACHINING_SCHEMA', 'MILLING_SCHEM
A'));
ENDSEC;

DATA;
#1 = PROJECT('EXECUTE EXAMPL E1', #2,(#4),$,$,$);
#2 = WORKPLAN('MAIN WORKPLAN',(#23),$,#8,$);
#4 = WORKPIECE('SIMPLE WORKPIECE', #6,0.010
#6 = MATERIAL('ST-50','STEEL',(#7));

#7 = PROPERTY_PARAMETER('E  = 200000 N/M2')
```

Figure 5: Architecture of machining process control system based on STEP-NC.

cutting force signal, which is acquired from dynamometer, and to adjust the feed rate. Interpolation algorithm is called to calculate the position of interpolation point according to the new feed rate in real-time. Servo system, which drives the feed shafts and spindle, is controlled via hardware interface to implement the adaptive control algorithm. The acquired cutting force data is saved in a data file and associated with corresponding working step by modifying the entities of input STEP-NC file and saved as output STEP-NC file.

4. Design and Validation of Fuzzy Control Algorithm

4.1. Fuzzy Control Algorithm with Self-Adjusting Factor for Constant Force Milling. The complexity of cutting process makes it hard to be modeled accurately with state-space equations. Artificial intelligence algorithms, which are able to handle unpredictable, nonlinear, multivariable, and incertitude controlled objects, can be a feasible solution. However,

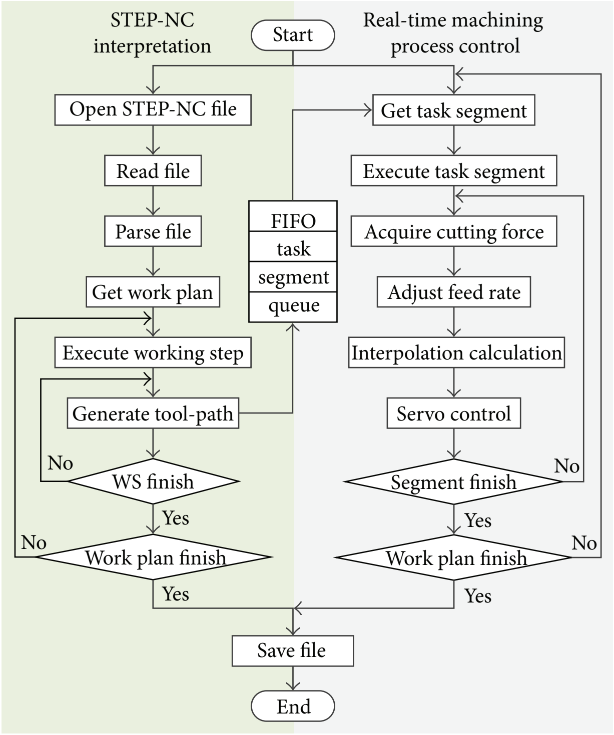


Figure 6: Procedure of real-time process control based on STEP-NC.

most artificial intelligence algorithms are used for off-line machining process optimization due to the limitation of computational complexity. Fuzzy logical algorithm, which has high computational efficiency, is suitable for real-time machining process control. In this paper, a fuzzy control algorithm with self-adjusting factor is proposed to adjust the feed rate in real-time in order to get constant cutting force. The principle of proposed control algorithm, which is designed by modifying the conventional fuzzy control algorithm, is illustrated in Figure 7. The input of control system is ideal cutting force while the controlled object is actual cutting force. The input lingual variables of fuzzy controller are the error between ideal cutting force and actual cutting force and the variety rate of error. The output lingual variable of fuzzy controller is the variety rate of feed rate. CNC system adjusts the feed rate according to the output quantity of fuzzy controller.

The error and error variety rate of ideal cutting force and actual cutting force can be calculated as

$$e = F_{\mathrm{id}} - F, \qquad \dot{e} = \frac{de}{dt}, \tag{2}$$

where F_{id} is ideal cutting force (N), F is actual cutting force (N), e is error (N), and $\dot{e}$ is error variety rate (N/s).

Then the error and error variety rate are quantified according to the domain of fuzzy rule input as

$$E = K_e e, \qquad \dot{E} = K_{ed}\dot{e}, \qquad K_e = \frac{3}{k_e F_{\mathrm{id}}}, \qquad K_{ed} = \frac{3}{k_{ed} F_{\mathrm{id}}}, \tag{3}$$

where E is fuzzy variable for error, $\dot{E}$ is fuzzy variable for error variety rate, K_e is quantitative factor of error (N^{-1}), K_{ed} is quantitative factor of error variety rate (s/N), k_e is adjusting factor of K_e, and k_{ed} is adjusting factor of K_{ed} (s^{-1}).

The quantitative factors K_e and K_{ed} are associated with ideal cutting force in order to improve the adaptability of fuzzy controller. As shown in Figure 8, the domain of fuzzy rule input is $[-3, 3]$ and triangle function, which has been widely used and has low computational complexity, is adopted as membership function of fuzzy rule. The fuzzy language value and membership value of input are expressed as follows:

A_1 and A_2 are fuzzy language values of error.

B_1 and B_2 are fuzzy language values of error variety rate.

$\mu_{A_1}(E)$ and $\mu_{A_2}(E)$ are membership value of error.

$\mu_{B_1}(\dot{E})$ and $\mu_{B_2}(\dot{E})$ are membership value of error variety rate.

The output fuzzy language values C_1 and C_2 are calculated by using A_1, B_1, A_2, and B_2 as

$$C = -\left[\alpha A + (1-\alpha) B\right], \qquad \alpha = \frac{1}{3}\left(\alpha_s - \alpha_0\right)|A| + \alpha_0, \qquad 0 \le \alpha_0 \le \alpha_s \le 1, \tag{4}$$

where α is self-adjusting factor, α_s is upper limit of self-adjusting factor, and α_0 is lower limit of self-adjusting factor.

The self-adjusting factor α is used to adjust the weight of error and error variety rate adaptively when calculating the output fuzzy language value. The upper and lower limits of self-adjusting factor are set to 0.85 and 0.45. The fuzzy control output and variety rate of feed rate are calculated as

$$U = \frac{C_1\mu_{A_1}(E)\,\mu_{B_1}(\dot{E}) + C_2\mu_{A_2}(E)\,\mu_{B_2}(\dot{E})}{\mu_{A_1}(E)\,\mu_{B_1}(\dot{E}) + \mu_{A_2}(E)\,\mu_{B_2}(\dot{E})}, \qquad \Delta f = K_f U, \quad K_f = \frac{k_f}{3}, \tag{5}$$

where U is fuzzy control output, Δf is variety rate of feed rate ($\mathrm{mm/s^2}$), K_f is scaling factor of fuzzy control output ($\mathrm{mm/s^2}$), and k_f is adjusting factor of K_f ($\mathrm{mm/s^2}$).

4.2. Test of Fuzzy Control Algorithm. Factors k_e, k_{ed}, and k_f can be used to adjust the rapidity, accuracy, and stability of fuzzy control system. Lower k_e and k_f can improve the rapidity but reduce the accuracy and stability. Lower k_{ed}

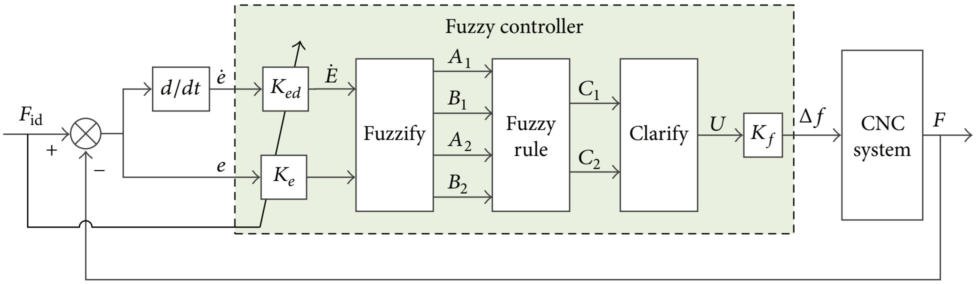


FIGURE 7: Constant milling force control algorithm based on fuzzy logic.

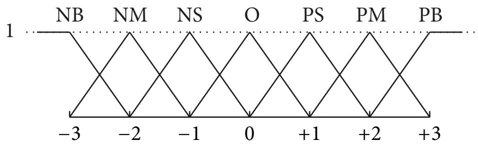
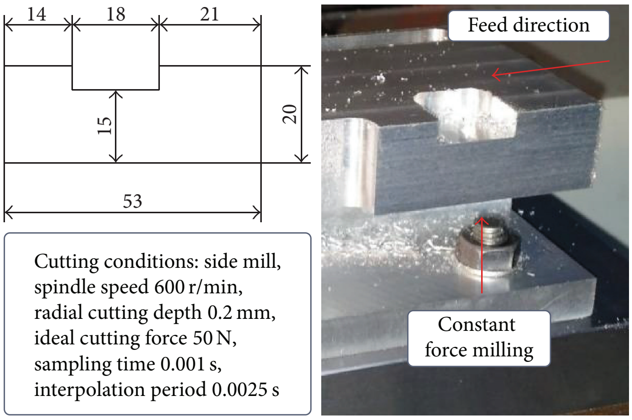


FIGURE 8: Membership function of fuzzy rule.



Cutting conditions: side mill,
spindle speed 600 r/min,
radial cutting depth 0.2 mm,
ideal cutting force 50 N,
sampling time 0.001 s,
interpolation period 0.0025 s

FIGURE 9: Experimental part for validation of fuzzy control algorithm.

can improve the stability but the rapidity is reduced. The proposed fuzzy control algorithm is validated and adjusted by actual cutting tests as shown in Figure 9. The test part is made of aluminum alloy and the cutting tool used here is a high speed steel end mill tool that is 10 mm in diameter and has 3 teeth. Considering the inertance of feed shaft, the feed rate is adjusted every 10 interpolation periods. The cutting force data is firstly processed by using mean filter algorithm as

$$F_k = \frac{1}{m}\sum_{i=k-m}^{k} F_{ci}, \quad m = \left[\frac{60}{n \times t_c}\right], \tag{6}$$

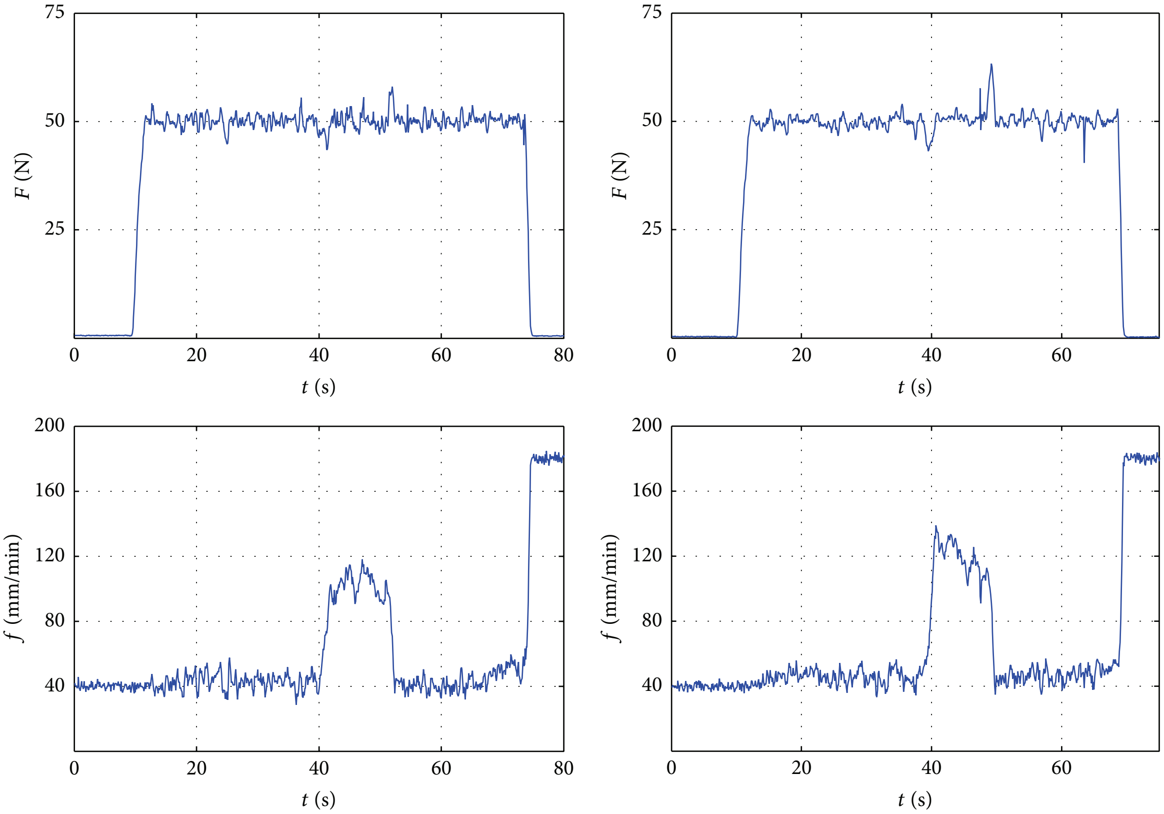
where F_k is average cutting force at the kth acquisition cycle (N), F_{ci} is actual cutting force at the ith acquisition cycle (N), n is spindle speed (r/min), and t_c is sampling time of cutting force (s).

The test results are shown in Figure 10, in which the subtitle is in format of "k_e-k_{ed}-k_f." The feed rate fluctuates obviously when $k_f = 8.0\,\mathrm{mm/s^2}$ and more steadily when $k_f = 4.0\,\mathrm{mm/s^2}$. The change of cutting force signal is more smooth when $k_{ed} = 7.2\,\mathrm{s^{-1}}$. So in this paper, adjusting factors k_e, k_{ed}, and k_f are set to 0.9, $7.2\,\mathrm{s^{-1}}$, and $4.0\,\mathrm{mm/s^2}$. The results indicate the fuzzy control algorithms proposed in this paper can work properly without building the model of metal cutting process with state-space equations.

5. Experimentation

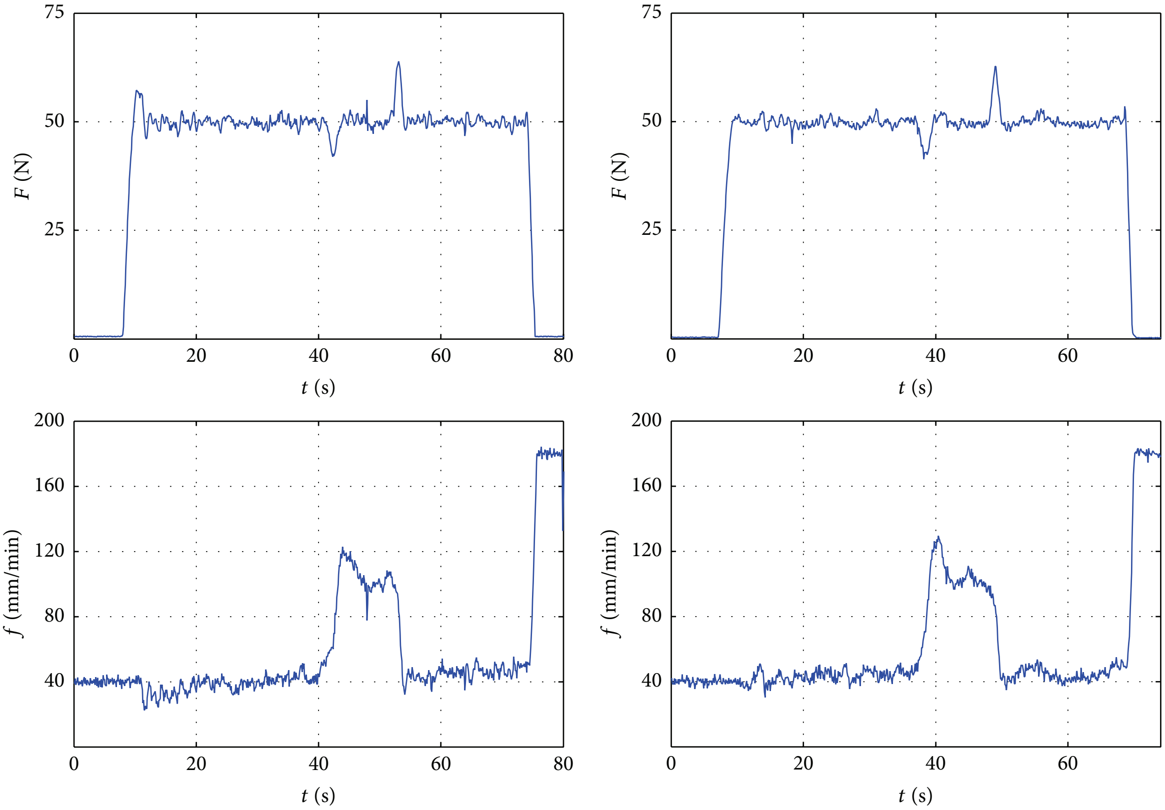
Experiments are designed and carried out to verify the function of proposed STEP-NC controller for real-time machining process control. The experiment platform consists of Qier XKV715 vertical milling machine, Kistler 9257B dynamometer, and industrial computer is shown in Figure 11. The proposed software CNC kernel of STEP-NC controller along with real-time machining process control algorithm runs on an industrial computer, in which a Rexroth PCM-S11.2 SERCOS master card is used for communicating with servo drivers and IO ports, and an Advantech PCI-1741U data acquisition card is used for collecting cutting force data from dynamometer. Both expansion cards are plugged in the PCI slots of industrial computer.

Two test parts made of aluminum alloy are machined on the experiment platform as shown in Figure 12. Planer face is chosen as manufacturing feature and side milling is chosen as machining operation. High speed steel end mill tool that is 10 mm in diameter is used to machine the parts with side milling method. The planar face of test part 1 has six sections that are different in width, which will cause irregularly variation of axial cutting depth in machining process. The shape of planar face in part 2 is a right-angled trapezoid, which will cause linear variation of axial cutting depth in machining process. Taking test part 1 as an example, instances of key entities related to real-time machining process control are extracted from Part 21 STEP-NC file, which is interpreted, modified, and outputted by STEP-NC controller. As shown in Box 1, instance in line #23 connects with planar face in line #29 and side finish milling operation in line #45 is the machining working step to be executed in adaptive mode. Milling technology in line #50 represents the cutting parameters, and in its attribute list, a constant force milling adaptive control strategy is assigned. As represented in line #53, the ideal cutting force is 50 N in the minus Z direction of feature coordinate system. The STEP-NC controller interprets the STEP-NC file, generates



(a) 0.9-10.8-8

(b) 0.9-7.2-8

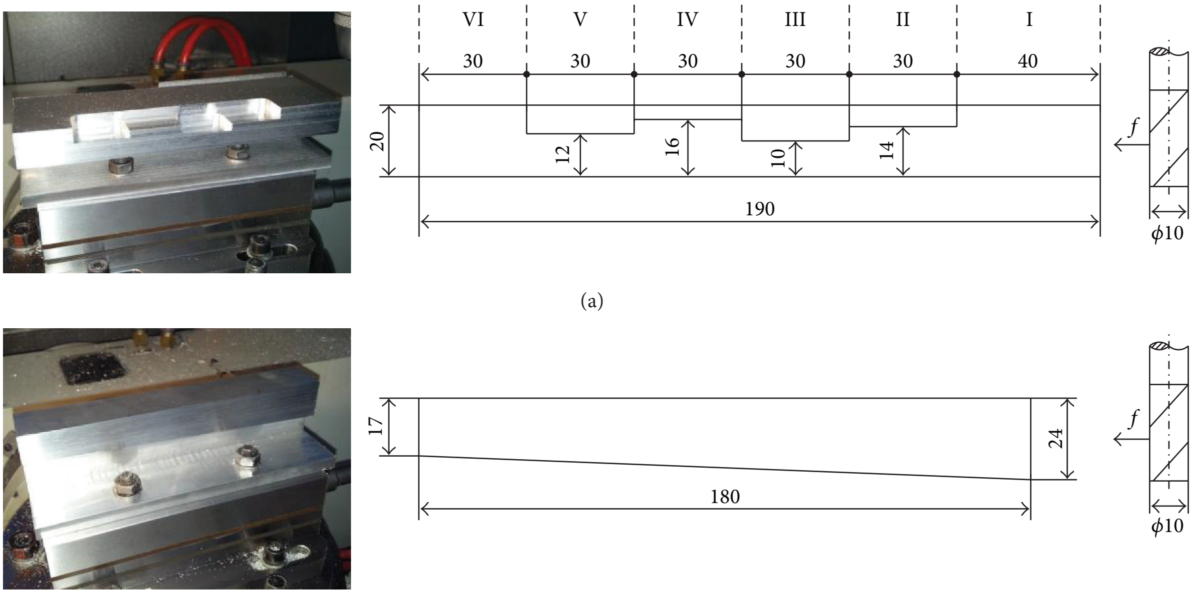


(c) 0.9-10.8-4

(d) 0.9-7.2-4

Figure 10: Selection of adjusting factors.

Figure 11: Experiment platform.

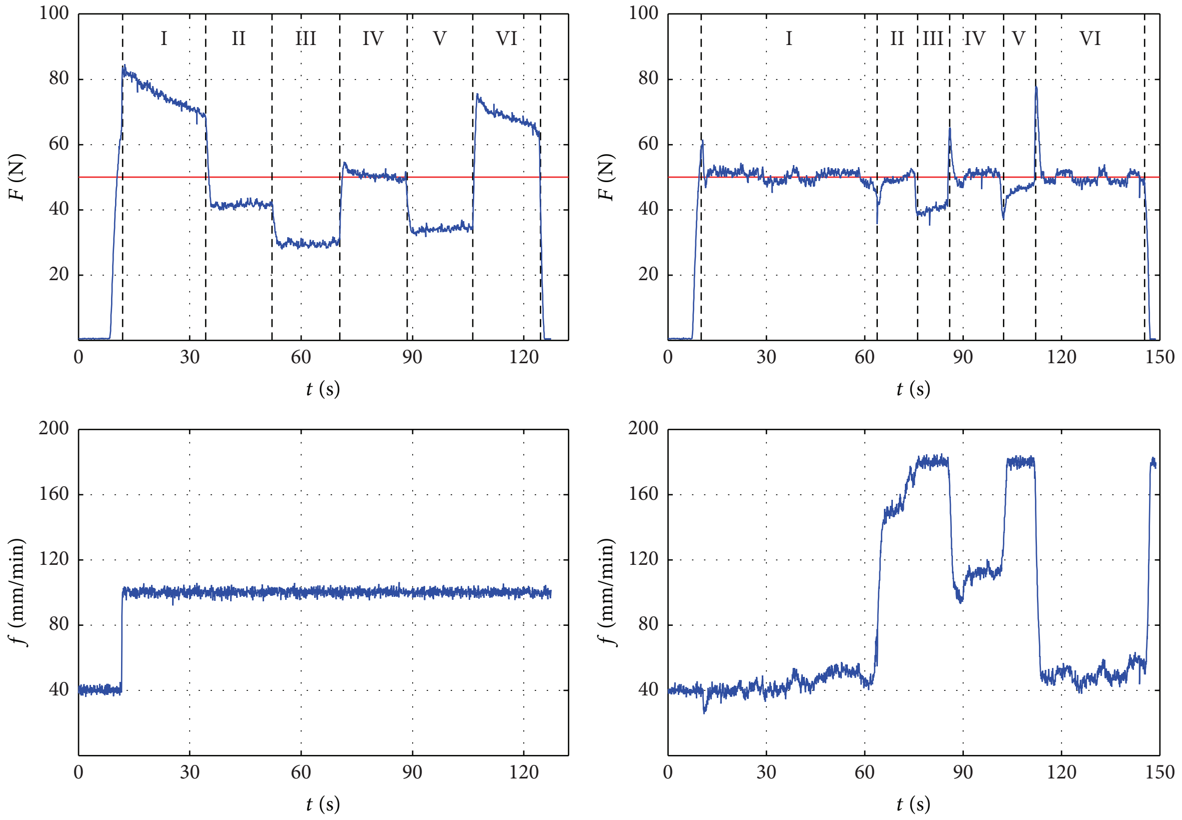


(a)

(b)

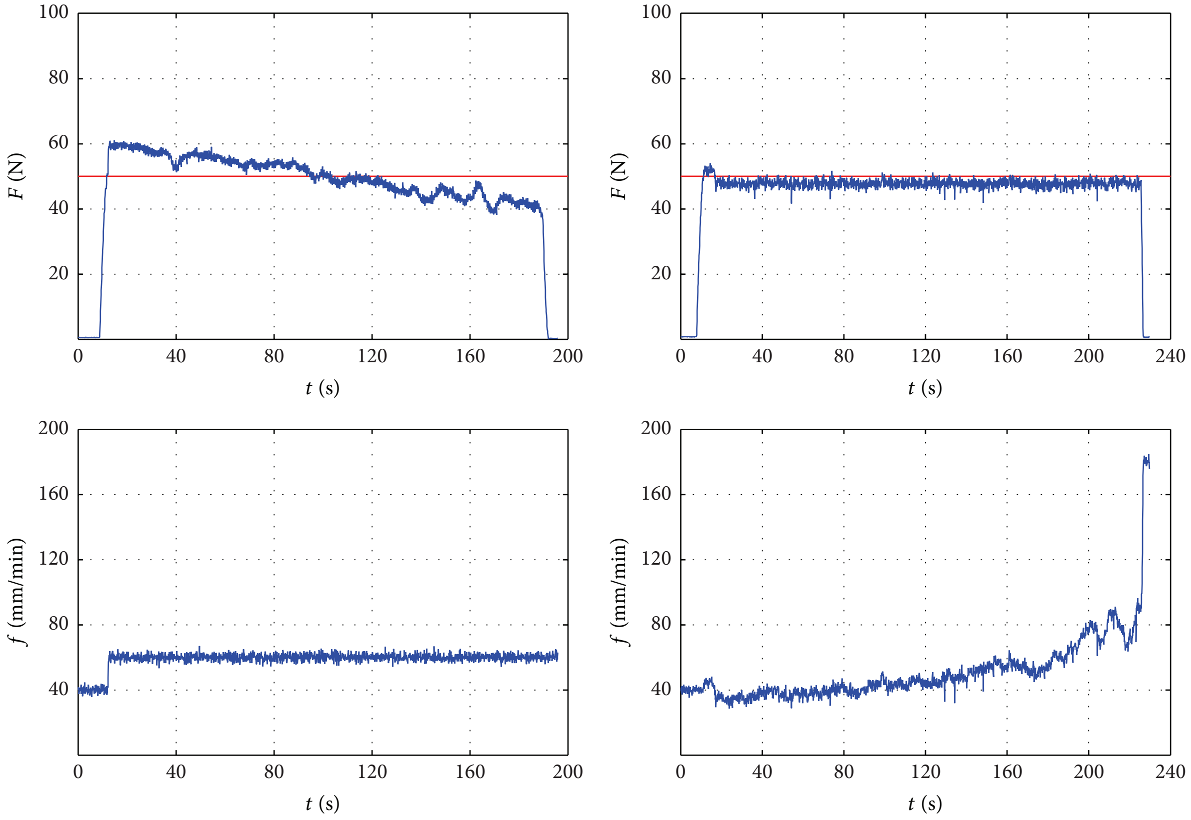
Figure 12: Parts and cutting condition of experiments. (a) *Cutting conditions*: side mill, spindle speed 1000 r/min, and radial cutting depth 0.5 mm. *Conventional machining*: feed rate 100 mm/min, result in Figure 13(a). *Constant force milling*: ideal cutting force 50 N, result in Figure 13(b). (b) *Cutting conditions*: side mill, spindle speed 2500 r/min, and radial cutting depth 0.5 mm. *Conventional machining*: feed rate 60 mm/min, result in Figure 13(c). *Constant force milling*: ideal cutting force 50 N, result in Figure 13(d).

tool path for each working step, and executes it in real-time. The acquired cutting force signal is saved in a data file that is stored at local hard disk after a working step is finished. Lines #54 to #56 are added in the input file by STEP-NC controller to preserve the information of real-time machining process condition. Line #56 is the newly added instance of entity *milling_force_save*, and in its attribute list, there are the path of cutting force data file and corresponding working step represented in line #23, which can be linked with each other. Lines #54 and #55 are instances of *cartesian_point*, which represent the position of milling cutter in feature coordinate system when the data acquisition is at start and end.



(a) Conventional machining

(b) Constant force milling



(c) Conventional machining

(d) Constant force milling

FIGURE 13: Results of real-time machining process control.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('STEP-NC EXAMPLE','A PART WITH SIDE PLANAR_FACE'),'1');
FILE_NAME('SIDEMILL_CF.STP','2015-05-16',('HU PO'),('RDNCT, HIT'),$,'ISO 14649',$);
FILE_SCHEMA(('MACHINING_SCHEMA','MILLING_SCHEMA'));
ENDSEC;
DATA;
...
/* * * * * * * * * * * * * * * * * * */
/* Instances of input STEP-NC file */
/* * * * * * * * * * * * * * * * * * */
#23 = MACHINING_WORKINGSTEP('WS FINISH PLANAR FACE1',#24,#29,#45,$);
#29 = PLANAR_FACE('PLANAR FACE1',#4,(#45),#30,#34,#39,#43,$,());
#45 = SIDE_FINISH_MILLING($,$,'FINISH PLANAR FACE1',8.0,$,#46,#50,#51,$,$,$,$,$,$,$);
#50 = MILLING_TECHNOLOGY(100.0,.TCP.,$,-1000.0,$,.F.,.F.,.F.,#53);
#53 = CONST_FORCE_MILLING(.F.,.F.,.T.,$,$,-50);
/* * * * * * * * * * * * * * * * * * * * * * */
/* Instances added by STEP-NC controller */
/* * * * * * * * * * * * * * * * * * * * * * */
#54 = CARTESIAN_POINT('START POINT',(21.0,-10.0,4.5));
#55 = CARTESIAN_POINT('END POINT',(21.0,200.0,4.5));
#56 = MILLING_FORCE_SAVE(#23,#54,#55,1000.0,'C:\\_OP\\Force_WS23-1.txt',$);
ENDSEC;
END-ISO-10303-21;
```

Box 1: STEP-NC file with real-time machining process control information.

6. Results and Discussion

Results of real-time machining process control are presented in Figure 13. The average cutting force per revolution and feed rate are represented as absolute value in the data graph, because all values of milling force and feed rate are negative according to the definition of feature coordination system. Approach feed rate is set to 40 mm/min for both conventional and constant force milling, and the adjusting range of feed rate in constant force milling algorithm is from 20 mm/min to 180 mm/min. For test part 1, cutting force signal is step-like while the feed rate is constant in conventional machining process as presented in Figure 13(a). In constant force milling process, cutting force signal changes suddenly at the start of every section while the proposed constant force milling algorithm decelerating or accelerating until the actual cutting force is equal to ideal cutting force or the feed rate reaches upper or lower limit as presented in Figure 13(b). For test part 2, cutting force signal changes gradually while the feed rate is constant in conventional machining process as presented in Figure 13(c). In constant force milling process, the feed rate is adjusted by the proposed fuzzy control algorithm and actual cutting force is kept close to ideal value as presented in Figure 13(d). Experiment results show that the STEP-NC interpreter can identify the ideal cutting force from STEP-NC file and keeps the actual cutting force around it by adjusting feed rate in real-time. The cutting force signals are recorded and linked with corresponding machining working step correctly. As a result, high-level information of real-time machining process control, instead of low-level information in most other published methods, can be interpreted at shop floor directly by using the proposed STEP-NC data model. Real-time machining process control algorithm is integrated with the software CNC kernel rather than external MPC in other research works.

7. Conclusion

This study has focused on the information exchanging mechanism, implementation method, and system scenario of real-time machining process control. An advanced solution method for optimization problems in manufacturing is proposed for implementing real-time machining process control at shop floor. Two key issues for implementing real-time machining process control are studied and solved. The issue of exchanging high-level product manufacturing data between CAPP systems and CNC systems is solved by extending STEP-NC standard and building an open STEP-NC controller that interpret STEP-NC data directly at shop floor. The issue of implementing real-time machining process control at shop floor is solved by integrating adaptive control algorithm with interpolation algorithm in software CNC kernel. Cutting force at specific direction is chosen to be the controlled object of real-time machining process control to demonstrate the newly proposed STEP-NC data model and implementation methods. A fuzzy control algorithm with self-adjusting factor is proposed to keep the cutting force constant by adjusting feed rate in real-time. The verification of proposed STEP-NC data model and implementation method is carried out on an open CNC platform. The experiment results indicate that the STEP-NC controller is able to interpret STEP-NC file with real-time machining process control functions correctly and keep the cutting force at specific direction close to ideal value. More research works on performance of real-time machining process control algorithms will be carried out based on the proposed implementation method in the future.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This study is financially supported by National Science and Technology Major Projects of China (Grant no. 2013ZX04013-011).

References

[1] L. Li, F. Liu, B. Chen, and C. B. Li, “Multi-objective optimization of cutting parameters in sculptured parts machining based on neural network,” *Journal of Intelligent Manufacturing*, vol. 26, no. 5, pp. 891–898, 2015.

[2] H. Huang, A. P. Li, and L. Y. Xu, “Fuzzy control of milling process with constraint of spindle power,” in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA ’08)*, vol. 1–23, pp. 7936–7939, Chongqing, China, June 2008.

[3] U. Zuperl, F. Cus, and M. Reibenschuh, “Neural control strategy of constant cutting force system in end milling,” *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 3, pp. 485–493, 2011.

[4] U. Zuperl and F. Cus, “System for off-line feedrate optimization and neural force control in end milling,” *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 2, pp. 105–123, 2012.

[5] E. Osaba, E. Onieva, F. Diaz, R. Carballedo, P. Lopez, and A. Perallos, “A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: a case study on routing problems,” *International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 46–56, 2015.

[6] R.-E. Precup, M. L. Tomescu, and S. Preitl, “Fuzzy logic control system stability analysis based on Lyapunov’s direct method,” *International Journal of Computers, Communications & Control*, vol. 4, no. 4, pp. 415–426, 2009.

[7] C. Qian, “Global attractivity of periodic solutions in a higher order difference equation,” *Applied Mathematics Letters*, vol. 26, no. 5, pp. 578–583, 2013.

[8] T. M. Guerra and L. Vermeiren, “LMI-based relaxed nonquadratic stabilization conditions for nonlinear systems in the Takagi-Sugeno’s form,” *Automatica*, vol. 40, no. 5, pp. 823–829, 2004.

[9] S. Kumar, A. Nassehi, S. T. Newman, R. D. Allen, and M. K. Tiwari, “Process control in CNC manufacturing for discrete components: a STEP-NC compliant framework,” *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 6, pp. 667–676, 2007.

[10] R. Laguionie, M. Rauch, and J.-Y. Hascoet, “Simulation and optimization in a multi-process environment using STEP-NC,” in *Proceedings of the IEEE International Conference on Control and Automation (ICCA ’09)*, pp. 2384–2391, Christchurch, New Zealand, December 2009.

[11] J. G. Campos and M. Hardwick, “Manufacturing traceability automation using features and nc-functions,” *International Journal of Computer Integrated Manufacturing*, vol. 22, no. 2, pp. 112–128, 2009.

[12] J. G. Campos and L. R. Miguez, “Standard process monitoring and traceability programming in collaborative CAD/CAM/CNC manufacturing scenarios,” *Computers in Industry*, vol. 62, no. 3, pp. 311–322, 2011.

[13] M. Wosnik, H. Rüdele, and P. Klemm, “Process-informed machining objects for closed-loop process planning,” *International Journal of Computer Integrated Manufacturing*, vol. 22, no. 11, pp. 1054–1064, 2009.

[14] S. Kumar, S. T. Newman, A. Nassehi et al., “An information model for process control on machine tools,” in *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology*, vol. 66 of *Advances in Intelligent and Soft Computing*, pp. 1565–1582, Springer, Berlin, Germany, 2010.

[15] F. Ridwan, X. Xu, and G. Y. Liu, “A framework for machining optimisation based on STEP-NC,” *Journal of Intelligent Manufacturing*, vol. 23, no. 3, pp. 423–441, 2012.

[16] F. Ridwan and X. Xu, “Advanced CNC system with in-process feed-rate optimisation,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 3, pp. 12–20, 2013.

[17] S. Engin and Y. Altintas, “Mechanics and dynamics of general milling cutters. Part I: helical end mills,” *International Journal of Machine Tools & Manufacture*, vol. 41, no. 15, pp. 2195–2212, 2001.

[18] ISO, “Industrial automation systems and integration—product data representation and exchange—part 11: description methods: the EXPRESS language reference manual,” ISO 10303-11:2004, 2004.

[19] ISO, “Data model for computerized numerical controllers,” Part 10: General Process Data, pp. 1–172, 2001.

[20] ISO, “Data model for computerized numerical controllers,” Part 11: Process data for milling,pp. 1–81, 2002.

[21] ISO, “Data model for computerized numerical controllers,” in *Part 111: Tools for Milling*, pp. 1–20, 2002.

[22] “iso-14649-toolkit,” 2013, http://code.google.com/p/iso-14649-toolkit/.