

XML 형식의 STEP-NC 파일로 구동되는 PC 기반의 STEP-NC milling machine

이원석*, 방영봉**

PC Based STEP-NC Milling Machine Operated by STEP-NC in XML Format

Wonseok Lee*, Young-Bong Bang**

ABSTRACT

Most of NC machines are operated by ISO6983 standard called G-code, which was developed in the early days of machine tools. This G-code limits hardware performance of the currently developed high-performance hardware & machine tools. By describing only movements of tool, almost all of information of previous production departments is lost, and the machining department cannot exchange information with other departments. For adjusting new hardware environment and direct communication of CNC machines with CAD/CAM software, ISO14649, STEP-NC is researched. This new standard stores CAD/CAM information as well as operation commands of CNC machines. In this research, the new CNC machine operated by STEP-NC was built and tested. Unlike other STEP-NC milling machines, this system uses the STEP-NC file in form of XML as data input. It makes possible for STEP-NC machines to exchange information to other databases using XML. The mentioned system of this paper loads the XML file, analyzes it, makes tool paths of two5D features with information of STEP-NC, and machines automatically without making G-code. All of software is programmed with Visual C++, and the milling machine is made with table milling machine, step motors, and motion control board for PC that can be directly controlled by C++ commands. All modules of software and hardware were independent, it allows convenient for substitution and expansion of the milling machine. The example 1 in ISO14649-11 that had all information about geometry and machining and the example 2 that has only geometry and tool information were used to test automatic machining by the open-architecture milling machine.

Key Words : STEP-NC, XML, PC, milling machine(밀링 머신), open architecture(개방형 구조), tool path(공구 경로)

1. 서론

현재 대부분의 생산 단계에서는 급속한 하드웨어의 발전에 맞추어 다양한 소프트웨어가 개발 이용되고 있다. 그러나 유독 생산 부분, 특히 기계

가공에서는 CNC 공작기계 개발의 초창기에 제정된 G-code 라 불리는 ISO6983 표준의 명령어 체계가 그대로 이용되고 있으며 모든 구동 명령 소프트웨어들이 이에 맞추어져 있다.

이 명령어들은 공작기계의 공구 축의 움직임

2002년 6월 15일 접수

* 서울대학교 기계항공공학부 대학원

** 서울대학교 기계항공공학부

을 기술하는 G-code 와 기계의 주변기능에 대해 기술하는 M 코드로 구성된다. 과거 하드웨어 성능이 부족하던 시기에 제정된 G-code 를 그대로 이용할 경우 다음과 같이 불리한 부분이 생긴다.

- 형상에 비해 명령어의 길이가 길어진다. 공구의 움직임은 기술하므로 간단한 직사각형의 형상을 기술한다 해도 그 길이가 길어진다.
- 직관적인 이해가 어렵다. 모든 명령이 G, M 의 코드와 숫자의 조합으로 기술되므로 그 의미를 파악하기 어렵다. 단순히 공구의 움직임만을 아는 수준을 넘어서, 형상이나 동작의 의미를 파악할 수 없다.
- G-code 를 만드는 과정에서 대부분의 CAD, CAM 정보가 누락된다. 생산의 이전 단계에서 만들어진 거의 모든 데이터가 소실되고 단지 축의 동작만이 남게 되며, 이로 인해 다른 생산 부분과 연계하기 어려워진다.
- 공작기계 메이커 간에 G-code 의 호환성이 부족하다. 앞서 언급한 문제들을 해결하고 새로운 기계 성능을 이용하기 위해 추가된 G-code 들은 공작기계 메이커 간에 차이를 보인다. 이로 인해 메이커 간에 호환성이 문제가 되며, 기계 선택에 제약이 생긴다.

이를 해결하고 새로운 기계 환경을 이용하기 위해 새로운 표준이 요구되는데, 현재 ISO14649 STEP-NC 가 준비되고 있다. 앞서 언급한 G-code 의 문제점을 해결하고, 3 차원 CAD 라는 새로운 환경에 대응하기 위해 제안된 STEP-NC 는 현재 F-dis 버전이 통과되어 국제 표준으로 정의되기 직전에 와있다^[1]. STEP-NC 는 단순한 기계의 동작 명령어 체계를 뛰어넘어 생산의 전 과정에서 발생하는 정보를 담고 있다. STEP-ISO10303 을 기반으로 하는 형상 정보에 기계 가공에 필요한 CAM 결과와 기계 구동명령어를 더하는 형태로 되어 있으며, 이로 인해 G-code 와 달리 정보의 누락이 발생하지 않는다. STEP 은 3 차원 형상을 기술하는 CAD 표준으로 이 표준을 바로 이용하기에 STEP-NC 는 3 차원 형상을 그대로 기술하여 이용할 수 있으며, CAM 정보가 형상과 함께 기술되므로 공구 경로 중 단순한 부분은 생략하고 자동으로 기계가 공구 경로를 작성하여 가공할 수 있게 된다^[2]. 이로 인해 도면을 그리고 별도의 프로그램으로

공구 경로를 생성한 후 이를 기계에 입력하여 가공하는 기존의 방식이 하나의 STEP-NC 프로그램 내에서 공구경로의 해석과 가공이 동시에 수행되는 새로운 방식으로 변하게 된다. 즉, STEP-NC 머신은 모든 생산단계의 정보를 지니고 이를 이용하여 자동으로 가공을 수행하는 능력을 가지며, 필요한 경우 외부의 생산 부서와 직접 자료를 주고 받을 수 있게 된다. 이를 위해서는 기존의 명령어 체계와 완전히 다른 STEP-NC 를 처리하는 소프트웨어와 함께 G-code 가 배제된 명령어를 따라 가공을 수행하는 하드웨어가 필요해진다^[3].

이러한 소프트웨어와 하드웨어에 대한 연구는 다양하게 수행되고 있다. 독일과 미국^[4]에서 STEP-NC machine 이 개발되어 시연회를 행한 바 있으며, 국내에서도 POSTEC^[5]에서 prototype 을 공개했다. 이 machine 들의 경우 STEP-NC 를 그대로 입력 파일로 이용하였으며, 이로 인해 공작기계가 정보를 바로 공유 할 수 있는 한계를 생산부서만으로 제약하는 결과를 만들었다. 본 논문에서의 machine 은 입력용 STEP-NC 파일의 내용을 XML 형식으로 변환하여 이 XML 파일을 이용함으로써 생산 부문을 넘어 XML 을 기반으로 하는 다른 데이터베이스 들과 정보를 바로 공유할 수 있는 기반을 제시하였다. 본 논문에서는 이러한 XML 을 이용하는 STEP-NC milling machine 을 제작하고 이를 테스트한 결과에 대해 다룬다. 이 STEP-NC milling machine 의 제작을 위해 상용의 수동 밀링 머신에 스텝모터를 부착하여 CNC 화하였고, 소프트웨어 부분은 모두 Visual C++과 MSXML 을 이용하여 자체 제작하였다. 모든 시스템은 Microsoft 의 Windows98 과 Windows NT 및 2000 환경에서 테스트되었다.

2. STEP-NC 파일의 내용

STEP-NC 는 앞서 언급한 대로 제품의 생산단계에서 발생하는 거의 모든 데이터를 저장한다.

STEP-NC 의 경우 자료의 서술이 인간의 직관적이 이해 위주로 되어 있어 기계가 바로 처리하기 어렵고, 공작기계에 바로 이용되는 공구 중심의 움직임이 아닌 가공의 필요한 모든 정보를 포함하므로 용량이 상대적으로 커지지만, 현재의 현장에서 G-code 외에 작업 매뉴얼, 제품 도면 등

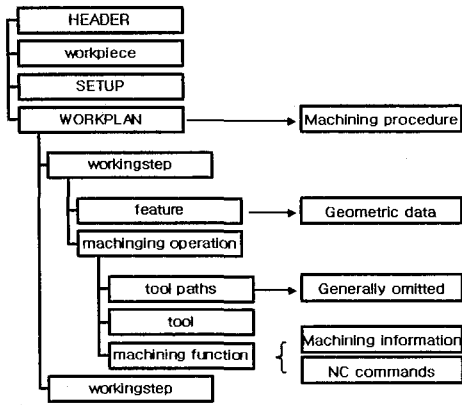


Fig. 1 Example of STEP-NC, ISO 14649-11 example 1

```

<main_workplan name="WORKPLAN" id="#2">
  <its_id name="identifier">"MAIN WORKPLAN"</its_id>
  <its_elements name="MACHINING_WORKINGSTEP" id="#10">
  <its_elements name="MACHINING_WORKINGSTEP" id="#11">
  <its_elements name="MACHINING_WORKINGSTEP" id="#12">
    <its_id name="identifier">"WS REAM HOLE1"</its_id>
    <its_section name="ELEMENTARY_SURFACE" id="#62">
    <its_feature name="ROUND_HOLE" id="#17">
    <its_id name="identifier">"HOLE1 D=22MM"</its_id>
    <its_workpiece name="WORKPIECE" id="#4">
    <feature_placement name="AXIS2_PLACEMENT_3D" id="#81">
    <depth name="ELEMENTARY_SURFACE" id="#64">
    <diameter name="TOLERANCED_LENGTH_MEASURE" id="#58">
    <change_in_diameter name="taper_select"/>
    <bottom_condition name="THROUGH_BOTTOM_CONDITION" id="#26"/>
    </its_feature>
    <its_operation name="REAMING" id="#21">
    <its_effect name="in_process_geometry"/>
    </its_elements>
    <its_elements name="MACHINING_WORKINGSTEP" id="#13">
    <its_elements name="MACHINING_WORKINGSTEP" id="#14">
    <its_channel name="channel"/>
    <its_setup name="SETUP" id="#8">
    <its_effect name="in_process_geometry"/>
    </main_workplan>
  <its_workpieces name="WORKPIECE" id="#4">
  <its_owner name="person_and_address"/>
  <its_release name="date_and_time"/>
  <its_status name="approval"/>
</PROJECT>
</STEP-XML>

```

Fig. 2 The part of XML file used for data input

이 추가로 이용되고 있다는 점을 고려하면, 용량에서 불리한 점이 있다고는 볼 수 없다. 현재의 발달된 하드웨어 기술로 인해 거의 모든 소프트웨어가 사용자 쪽에 가깝게 발달하는 것을 고려하면, 단순한 용량이나 처리 속도의 비교보다는 이용자의 편의와 정보의 통합이 가능한 STEP-NC 쪽에서 많은 유리한 점이 확인된다. CAD, CAM, CNC의 정보를 유기적으로 결합하여 저장하기 위해 STEP-NC는 workingstep이라는 단위 별로 자료를 묶어서 저장하며 이 workingstep의 관계를 WORKPLAN에서 설명한다. STEP-NC의 형식은 Fig. 1과 같다. 가공을 하고자 하는 대상을 workpiece에서 정의한다. 이 workpiece와 공작기계의 초기화에 대한 내용을 setup에서 정의하여 가공을 위한 기본적인 준비를 마친다. 가공 대상의 형상은 feature라고 하는 하나의 완결된 단위 형상별로 정의하며, 이 feature에서 하나의 의미를 지니고 하나의 공구로 가공 가능한 부분을 workingstep으로 정의한다. 이 workingstep 안에 공구와 CAM, CNC 명령을 담는다. CAM 부분에는 가공정보가 담겨 있으며, 이 정보와 feature의 형상정보를 이용하여 공구 경로를 만든다. 공구 경로 자체는 ISO14649-11에 선택사항으로 정의되어 있지만, 일반적으로 입력하지 않는다. CNC 명령에는 공작기계의 축 동작과는 무관한 기타 명령들이 담겨 있다. 이런 workingstep들이 어떤 순서로 가공되고 어떤 의미가 있는지를 기술하는 부분이 WORKPLAN이며, 이 WORKPLAN을 따라 모든 작업이 진행된다^[6].

본 연구에서는 STEP-NC 자체를 바로 사용하지 않고 XML로 변형하여 이용한다.

3. 입력용 XML 파일

본 논문의 STEP-NC milling machine은 다른 prototype들과 달리 STEP-NC 파일과 동일한 내용과 구조를 갖는 XML 파일을 입력에 이용한다^[7].

XML 파일은 내부적으로 tree 구조를 가지고 있으므로, STEP-NC의 상속관계를 바로 적용할 수 있으며, tag명에 STEP-NC entity명을 그대로 이용할 수 있으므로 STEP-NC와 동일한 내용과 구조를 갖는 XML 파일을 만들 수 있게 된다. 또한 tag명 뒤의 참조 항목에 원래의 STEP-NC에서의 행번호를 기입하고, subtype명을 명기함으로써, C의 함수형식을 빌려 정의하는 STEP-NC보다 이해하기 쉽고, 별도의 참조 자료 없이 이용하기 쉽게 된다. 이러한 XML 파일의 내용은 Fig. 2와 같다. 이 부분은 뒤에서 이용할 예제 1 중, hole 형상을 정의하는 부분이다. 본 논문에서는 이러한 XML 파일을 다음과 같은 이유로 이용하였다.

- XML 파일을 다루는 많은 방법들과 도구들이 존재한다.
- STEP-NC 정보를 다루기 위한 별도의 자료구조가 필요하지 않다.
- 다른 전자 상거래에서 XML을 표준으로 이용하므로 현재의 시스템을 인터넷 환경에 적용하기 유리하다.
- 현재 여러 분야의 데이터베이스에서 XML을 이용하므로 타 부서와 자료 교환이 쉽다.
- Microsoft와 IBM에서 XML을 지지한다.

현재의 시스템의 다른 부분들을 Visual C++로 작업하였으므로, XML 을 다루기 위한 도구로 Microsoft 에서 제공하는 Visual C++ 용 XML parser 인 MSXML 을 이용하였다.

4. 소프트웨어 부분의 제작

본 논문의 시스템의 소프트웨어 부분은 크게 XML 을 해석하고 이 내용을 따라 각각의 가공을 명령하는 부분과 각각의 형상을 가공하는 부분, 그리고 이 정보들을 이용하여 기계를 구동하는 부분으로 나뉜다. 본 논문에서는 모든 부분을 Microsoft 의 Visual C++로 자체 제작하였으며, 각각의 모듈의 세부적인 내용은 다음과 같다.

4.1 Main program

이 부분은 XML 을 읽어 그 내용을 따라 각각의 모듈을 가동시키고 가공 명령을 내리는 부분이다. XML 파일을 처리하기 위해 MSXML 을 이용하여 XML 파일을 Visual C++상에서 parsing 하고 필요한 자료를 추출하였다.

앞서 언급한 대로 WORKPLAN 을 찾아 그 하위에 위치하는 workingstep 을 순서대로 읽어 들이고 각각의 workingstep 에 기술된 feature 의 공구 경로 생성기를 호출한다. 동시에 feature 별로 가공에 필요한 정보를 읽어 들여, 필요한 경우 가공 정보 생성기를 이용하여 정보를 변경하거나 추가한다. 이 부분에서 읽어 들여 공구 경로 생성기에 전달하는 정보는 다음과 같다.

- 형상정보: feature 의 종류, 외곽선, 깊이, 위치와 자세, boss 의 유무 및 형상
- 가공정보: 절삭속도, 공구 회전수, 공구의 형상
- NC variable: 절삭유의 유무, 문의 개폐, 공구 교환 등의 기계 관련 명령 일체

정보가 부족하여 내부적으로 생성해야 할 경우를 위해 가공물의 종류, 기계의 출력, 공구의 재질 등의 정보 역시 준비된다. 이 정보들을 이용하여 필요한 가공 정보는 내부적으로 생성되고 원래의 XML 파일에 업 데이트 된다. 이를 위해 내부적으로 다음과 같은 계산이 가능하다.

- workingstep 분할 기능: 필요한 경우 하나의 workingstep 을 2~3 개의 workingstep 으로 분할하여 가공을 수행한다.

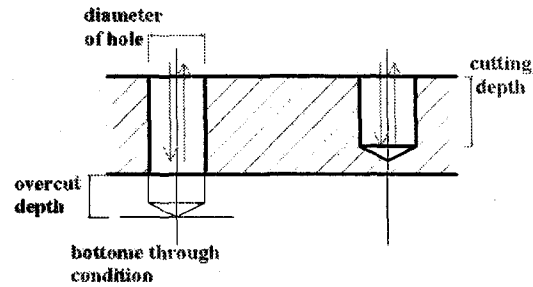


Fig. 3 Geometric information of hole features

- 절삭속도 및 공구 회전수 계산 기능: 가공물의 단위 부피 당 절삭에너지와 공구 정보, machine 의 출력을 이용하여 적절한 속도를 계산하며, 현재의 시스템은 형상정보를 가지고 있으므로, 내부적으로 최적의 값을 찾을 수 있다.

이런 정보가 준비되면, 주어진 형상별로 해당하는 공구 경로 생성기를 호출한다.

4.2 공구 경로 생성기

본 논문의 시스템은 two5D(2.5 차원의 ISO 14649 식 표현) 형상을 대상으로 하며 이러한 two5D 형상은 공구의 이동 방식에 따라 크게 두 가지로 나눌 수 있다. 가공물 표면에 대해 수직적 움직임만으로 절삭가공을 행하는 drilling type strategy 로 가공이 가능한 hole feature 와 수평적 움직임도 요구되는 milling type strategy 로 가공되는 그 외의 형상들- pocket, planar face, STEP, slot feature 들의 두 그룹은 공구 경로의 생성이나 이 때 필요한 정보에 있어서 큰 차이를 보인다. 현재의 공구 경로 생성기는 각각의 feature 에 대응하는 공구 경로 생성기 대신, milling type strategy 를 이용하는 feature 들을 가공하기 위한 공구 경로 생성기와 hole 에 관련된 workingstep 들을 처리하기 위한 공구 경로 생성기의 2 개의 공구 경로 생성기가 존재하며, 개별적인 형상이나 workingstep 은 이 기본틀에 맞춰서 변형된 후 처리된다. 두 공구 경로 생성기에 대한 설명은 다음과 같다.

4.2.1 Drilling type strategy

Hole feature 를 가공하기 위한 workingstep 들이 이에 속한다. Hole 은 Fig. 3 과 같은 정보를 갖는다.

Boring 을 제외한 모든 홀 가공은 동일한 방식

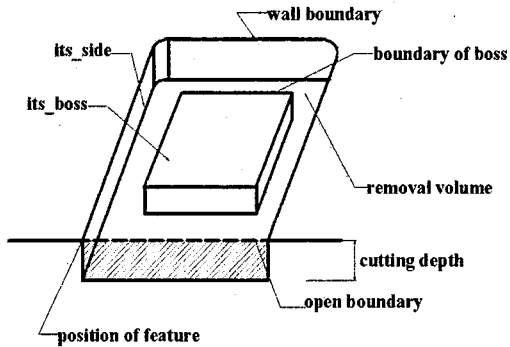


Fig. 4 General geometry of two5D feature of milling type strategy

으로 행해진다. 공구를 홀이 있는 위치로 이동시킨 후 표면에 수직하게 파고들어가며 가공을 행한다. 홀은 바닥면 조건에 따라 Fig. 3 처럼 주어지는 정보가 다르다. 관통 조건의 경우 홀의 깊이는 가공물의 두께와 동일해 지지만, 공구의 절삭깊이는 가공물의 두께에 과절삭 깊이만큼 더해진 값이 된다. 홀의 경우 홀의 직경은 별도로 표기하지 않으며, 공구의 직경과 동일하다고 가정한다. 깊은 홀의 가공을 위해 다단계 절삭을 할 경우는 1 회 절삭량과 후퇴량을 표기한다.

4.2.2 Milling type strategy

Milling 작업으로 가공 가능한 일반적인 two5D의 형상은 Fig. 4 와 같다. 주위의 미절삭 영역과 경계를 기점으로 나뉘어지며, 그 내부는 주위보다 절삭깊이 만큼 낮아진다. 이로 인해 옆면과 바닥면을 갖게 된다.

열린 경계의 경우 측면이 존재하지 않지만, 경계선은 존재하는 것으로 본다. 바닥면의 경우 관통 조건이 설정되면 존재하지 않을 수 있다. 경계선 내부의 재료는 절삭가공으로 제거되며, 경우에 따라 내부에 미 가공영역이 존재할 수 있으며, STEP-NC 에서는 이를 boss 라고 부른다. 이 boss 영역은 별도의 workingstep 으로 구별되지 않고 해당 feature 에 종속적으로 기술된다.

홀의 가공을 제외한 모든 two5D 형상은 이 기본 틀에 맞춰 변형시킬 수 있다. 따라서 이 형상에 대한 공구 경로 생성기를 제작하면, 대부분의 중복작업을 생략할 수 있다.

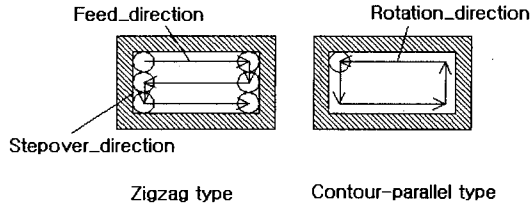


Fig. 5 Two types of tool paths of milling type strategy

Fig. 5 와 같이 two5D 의 형상을 가공하는 방법은 parallel 과 contour 의 두 가지 방식^[8]이 있으며 두 방식은 공통적으로 다음과 같은 정보를 가지고 있다.

- Offset 폭: 공구 경로간의 떨어진 거리
- Overlap 폭: 공구 경로간의 겹친 거리
- Tool 정보: tool 의 정보는 ISO14649-111 번의 방식을 따라 톨의 재료 및 형상 일체를 기술한다. 이 중 공구 직경은 offset 폭과 overlap 폭을 결정한다. 또한 공구 날의 길이는 한번에 절삭 가능한 깊이의 한계를 결정한다.
- 절삭 속도: 단위 회전 당 절삭량을 기술한다.
- 스피들 회전수: 공구의 회전 속도를 기술한다.
- 기계 동작 명령: 가공 중 필요한 기계 동작 명령들의 집합이다.

이들 공통 정보와는 별도로 절삭 방법마다 다른 정보를 갖는다. Contour 방식의 경우 공구가 외곽선을 따라 도는 방향과 안에서 밖인지 그 반대인지를 제공한다. Parallel 의 경우 절삭 방향과 이 공구 경로들의 offset 방향을 제공한다.

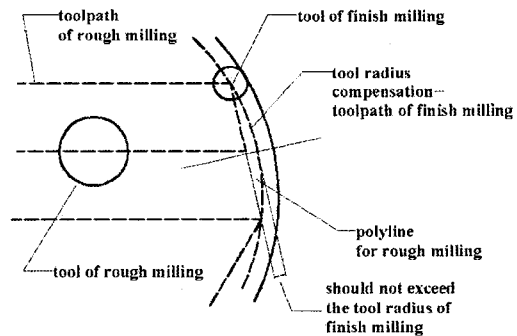


Fig. 6 Tool paths of rough milling and finish milling of the curved boundary feature

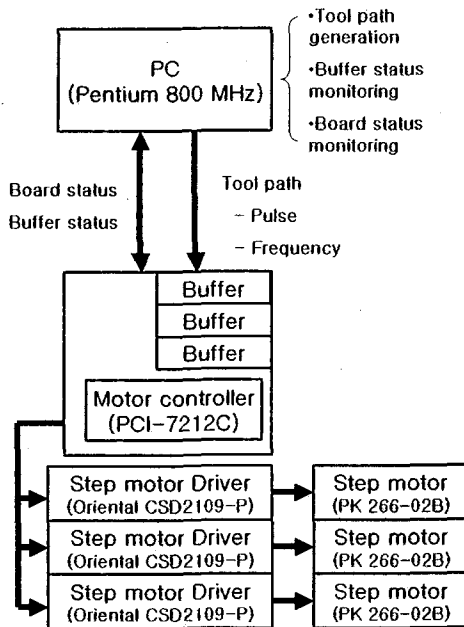


Fig. 7 Hardware components and its operation flow

이들 정보는 공구정보를 제외하고 전부 선택 사항이므로 공구 경로 생성기가 변형, 신조할 수 있다. 이렇게 변경된 정보는 원래의 XML 파일에 업데이트할 수 있다.

4.2.3 Feature with NURBS boundary

외곽선에 곡선이 있는 형상의 경우 황삭과 정삭을 구분하여 황삭의 경우 다각형으로 변형하여 처리한다. STEP-NC 에서는 정삭에서 tolerance 에 관해 표시하고 있다. 정삭의 공구 경로는 주어진 공차를 만족하도록 곡선을 자동으로 분할하여 단위 직선으로 보간하여 처리한다. 이 때 보간을 위한 모든 계산은 CPU 가 처리한다. 현재의 시스템은 하드웨어에서 분해능이 1/800mm 이므로 이를 공차로 이용하여 직선 분할을 수행하였다. 황삭의 경우는 Fig. 6 처럼 정삭에서 처리될 수 있을 정도의 오차를 갖는 다각형으로 변형하여 처리한다.

4.3 가공 명령 처리기

이 부분은 위의 부분들과 달리 독립적인 부분으로 존재하지는 않는다. STEP-NC 는 기계가 직접 공구 경로를 생성하고 이를 처리하는 구조이므로 공구 경로가 만들어지는 즉시 가공이 수행된다.

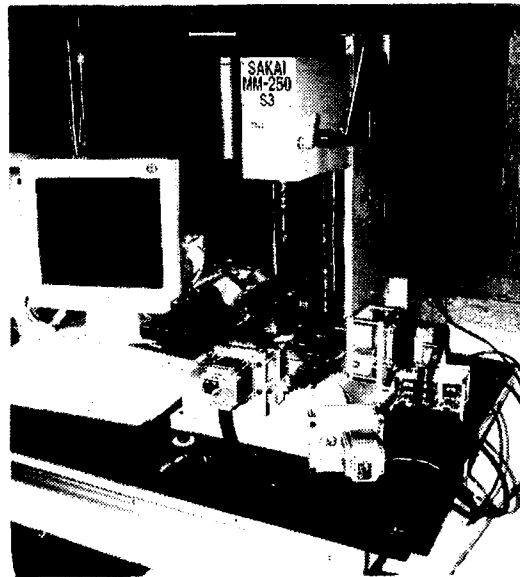


Fig. 8 STEP-NC milling machine of this research

일반적인 가공에서는 공구 경로의 생성이 완료된 후 공작기계의 작업이 수행되어 두 개의 부분은 서로 어느 한 쪽이 동작하면, 다른 한쪽은 대기하게 된다. 현재의 경우는 보드에서 지원하는 3 개의 버퍼를 확인하고 비는 버퍼에 공구 경로를 계산하여 입력하는 방식이므로, 공구 경로의 생성과 작업이 대기 없이 계속된다. 추후 리니어 모터가 이용된 고속공작기계에 적용하거나, 과거에 제작된 저사양 CPU 를 쓰는 공작기계에 적용할 경우를 고려하여, CPU 의 계산이 시스템에 가하는 부담을 줄이기 위한 방법으로 공구 경로의 생성 후 가공역시 가능하게 하였다. 곡선의 보간의 경우, 보드에서 처리하는 것 보다는 고속의 CPU 가 처리하는 것이 빠르므로 이 경우는 보다 유리할 수도 있다. 단위 공구 경로 생성 후 공작기계에 바로 전달하는 경우는 별도의 모듈이 아닌 공구 경로를 모터 동작으로 변환해 주는 서브 함수 형태로 존재하게 된다. 공구 경로 생성기가 만들어 낸 STEP-NC 공구 경로를 적절한 모터 동작으로 변환하는 부분이며, 이 부분을 변경하면, 하드웨어의 변동에 대처할 수 있게 된다.

5. 하드웨어 부분의 제작

현재의 상업용 공작기계 들은 거의 전부가 G-

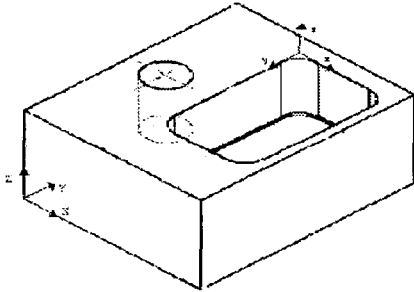


Fig. 9 ISO14649-11 example 1

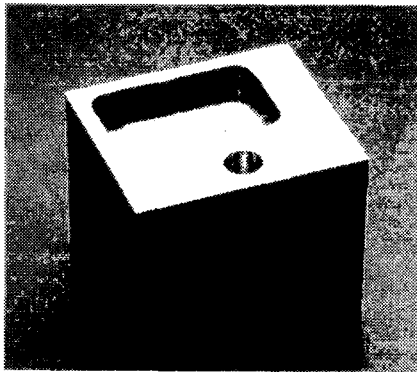


Fig. 10 Test result of example 1

code 를 이용하므로, 새로운 체계인 STEP-NC 시스템을 이들 기계로 구현하는 데에는 한계가 존재한다. 본 논문에서는 기존의 상업용 수동 밀링에 step motor 를 부착하여 CNC 화하였으며, 이 머신으로 STEP-NC milling machine 을 구현하였다. 부착된 step motor 의 동작을 위해 일본 interface 사의 PCI 7212C motion control board 를 이용하였다. 이 보드는 PCL6045 라는 모션컨트롤전용 LSI 를 사용하여 직선과 S 자 가감속방식의 펄스를 발생시킬 수 있도록 되어 있다. PC 의 소프트웨어 파트에서 해석하여 생성된 가공경로를 이 보드를 사용하여 모터의 동작으로 변환하고, 모든 가공 과정 중에 G-code 를 배제하였다. 이를 그림으로 나타내면 Fig. 7 과 같다. 현 시스템은 PC 와 모듈별로 제작된 파트를 이용함으로써 개방형 구조의 시스템을 구현하였으며, 제어기의 변경이 있을 경우 소프트웨어에서 가공 명령 처리부분을 변경하면, 어떠한 시스템에도 적용할 수 있게 된다. 만일 출력부분을 G-code

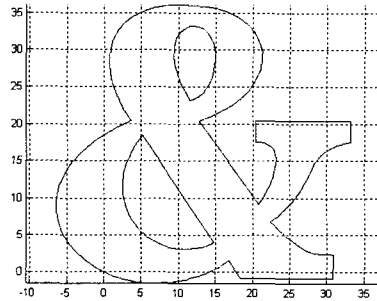


Fig. 11 Example 2 to test automatic machining under insufficient information

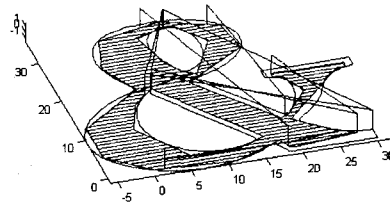


Fig. 12 All tool paths of example 2

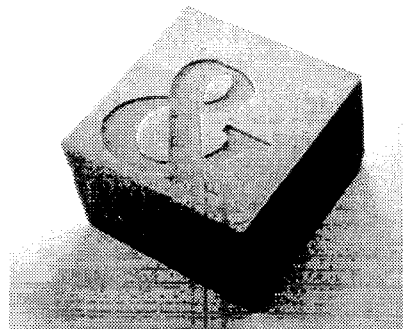


Fig. 13 Test result of example 2

의 text 파일로 할 경우는 기존의 CNC machine 에 그대로 이용하는 것도 가능하다. 제작된 시스템은 Fig.8 과 같다. 수동 밀링 머신은 일본 SAKAI 사의 MM-250 S3 를 이용하였고, 직교 3 축의 구동 모터로 오리엔탈모터사의 PK266 02B 를, 스텝모터 구동 드라이버로 정전류구동형 CSD2109-P 를 사용하였다. 스텝모터가 한 스텝 회전할 경우 테이블의 이송량은 1/800 mm 이다. 시스템은 현재 Windows98se 환경에서 동작하며, Windows NT 및 2000 환경에서도 테스트하였다.

6. 가공 테스트

제작한 machine 의 테스트를 위해 2 가지 경우에 대한 예제를 준비한다. 한가지 경우는 STEP-NC 파일이 완성되어 있는 경우로 필요한 모든 값들이 파일 내에 존재하는 경우에 대응하며, 또 하나의 경우는 STEP-NC 파일에 필요 최소한의 정보, 즉, 가공물의 재료, 공구, 형상만이 존재하는 경우에 대응한다. 전자의 경우를 위해 ISO14649 part 11 에 있는 예제 파일 1 번을 이용한다. 이 파일은 Fig. 9 와 같이 간단한 형상을 다루는 것으로 3 개의 feature 와 5 개의 workingstep 으로 구성되어 있으며, 필요한 모든 정보가 준비되어 있다. 이를 XML 파일로 변환하여 입력한다. 후자의 경우를 위해서 Fig. 11 과 같은 형상을 준비하였다. 이 형상은 내부에 2 개의 boss 를 가지며, 외곽선이 직선과 곡선들의 조합으로 되어 있는 pocket 이며, 현재 외곽선의 정보와 깊이 정보, 위치 정보만을 가지고 있다. 이 파일을 입력하여 자동으로 workingstep 을 분할하고 공구 경로를 생성하여 가공을 수행하는 것을 확인한다.

일반적인 기계 가공의 경우, 공구 중심의 이동을 기술하는 공구 경로의 생성과 이를 기계에 입력하는 과정이 필요하지만, 현재의 테스트에서는 모든 자료는 XML 형태의 STEP-NC 파일로 기계에 바로 전달되며, STEP-NC milling machine 이 공구 경로의 생성과 가공을 동시에 수행한다.

전자의 실험 결과는 Fig. 10 과 같다. 후자의 경우는 자동으로 황삭과 정삭이 구분되어 가공되었으며, Fig. 12 는 이 결과인 전체 공구 경로이다. 2 단계로 나뉘어 가공이 진행되었고, 다각형으로 형상을 치환하여 그 다각형을 먼저 가공한 후 곡선을 처리하는 정삭 가공이 진행되었다. 최종적으로 가공된 결과는 Fig. 13 에 나타나 있다.

두 가공 모두 STEP-NC 파일을 이용하여 전 가공 과정이 자동으로 수행 완료 되었다.

7. 결론

본 논문에서는 XML 형식의 STEP-NC 파일을 입력 받아 two5D 형상에 대해 자동으로 공구 경로를 생성하고 가공을 수행하는 STEP-NC 밀링머신의 개발과 그 테스트에 대해 다루었다. 본 연구에서 개발하고 테스트한 STEP-NC 밀링머신은 다

음과 같은 특징을 가진다.

- STEP-NC 를 이용하여 자동으로 공구 경로를 생성하고 가공을 수행한다.
- G-code 없이 모든 동작을 수행하며, 가공은 공구 경로의 해석과 동시에 수행된다.
- XML 을 이용하여 입력을 수행하므로 다른 XML 기반의 database 들과 자유롭게 자료를 교환할 수 있다.
- 주어진 STEP-NC 의 정보를 따라 충실히 가공을 수행하며, 부족한 정보가 있을 경우 내부적으로 계산하고 이 값을 이용하여 자동으로 가공을 행한다.
- Two5D 형상에 대해 자동으로 가공을 수행한다.
- PC 를 기반으로 하는 시스템으로 XML 을 이용하므로 network 환경에 적응하기 쉽다.
- PC 와 독립적인 모듈들을 결합하여 제작하였으며, 이로 인해 소프트웨어, 하드웨어 모두 각각의 부분들의 교환 및 확장이 자유로운 개방형 구조이다.

후 기

본 연구는 국제 IMS project 의 하나인 STEP-NC project 의 일환으로 행해졌으며, 한국측 partner 인 생산기술연구원의 연구비 지원으로 수행되었습니다. 본 논문에서 이용한 XML 파일을 제작해 주신 동일한 project 에 소속된 최종호 교수님 연구실에 감사 드립니다.

참고문헌

1. International Standards Organization, ISO14649-1 DIS version, "Industrial automation systems and integration - Physical device control - Data model for Computerized Numerical Controllers, Overview and Fundamental Principle," 2000.
2. 한수영, 장병열, 조현보, 서석환, "STEP-NC 구현을 위한 ISO14649 의 역할 및 발전 방향," 한국정밀공학회지, 제 17 권, 제 5 호, pp. 15-24, 2000.
3. Sang-Uk Cheon, "Shop Floor Programming System for STEP-NC," Korea-Germany Workshop on STEP-NC Technology, pp. 72-93, 2001

4. STEP-TOOLS Inc, "<http://www.steptools.com>"
5. Jounghoon Jo, Seokhuan Seo, "Tool path generation based on feature of STEP-NC and its Renewal (In Korean)," Journal of Korean Society of CAD/CAM Engineering, Vol. 4, No. 4, pp. 295-311, 1999.
6. International Standards Organization, ISO14649-10 FDIS version, "Industrial automation systems and integration - Physical device control - Data model for Computerized Numerical Controllers," 2001.
7. 이원석, 방영봉, "PC 기반의 STEP-NC 밀링 머신의 개발," 2002 한국 CAD/CAM 학회 학술발표회 논문집, pp. 335-341, 2002.
8. S.C. Park, B.K. Choi, "Tool-path planning for direction-parallel area milling," Computer-Aided Design 32 17-25, 2000.