

IBM-PC를 이용한 CNC 선반용 한글 자동 프로그래밍  
시스템과 그래픽 디버거의 설계

고 명 삼, 김 규 식, 김 대 원, °성 광 제  
서울대학교 공과대학 제어계측공학과

A Design of a Korean Automatic Programming System and a  
Graphic Debugger for CNC Lathe using IBM-PC

Myoung Sam Ko, Gyu Sik Kim, Dae Won Kim, °Kwang Jae Sung  
Dept. of Control & Instrumentation Eng. College of Engineering, Seoul National Univ.

Abstract

Although the use of NC machine is increasing in modern industry, unfamiliarities of the NC program syntax do not allow us to program it easily. So, automatic programming systems, such as APT, COMPACT, CL, were developed to help the uninitiated to write NC program easily. In this research, Korean Automatic Programming System for 2-axis NC lathe is developed, by which NC program is easily programmed using Korean letters. In addition, Graphic Debugger for 2-axis NC lathe is developed for debugging the NC program error and animating the cutting process. The above systems are run on the IBM-PC/XT.

1. 서론

기계가공의 자동화시스템은 다종다류의 기계가공물에 대하여 오랜 기간에 걸쳐 경제적으로 사용할 수 있어야 하며, 그 이유때문에 FMS(Flexible Manufacturing System)라는 개념이 생겨났다. 이때 이 개념의 기본을 이루는 요소로 등장할 수 있는 것이 로봇과 CNC(Computerized Numeric Control) 공작기계라 할 수 있다. 특히 1952년 미국의 M.I.T.에서 처음 NC 밀링머신을 공제한 이후, NC 공작기계에 대한 연구는 그 발전을 거듭하였고 [1], 최근 컴퓨터의 발달과 더불어 NC 공작기계 제어부의 발전이 눈부시게 이루어져서, 사용자가 NC 가공에 대한 전문적인 지식이 없어도 화면상에 나타나는 가공도면과 대화에 의해서 모든 입력을 처리할 수 있는 시스템으로까지 진보하였다.

본 논문에서는 현재 상용되고 있는 IBM-PC를 이용하여, 사용자가 쉽게 사용할 수 있도록 정의한 상위단계의 한글 NC 전용언어를 해독하여 CNC 선반의 입력으로 이용 가능한 코드로 변환하는 프로세서(Processor)를 설계하였고, 이 프로세서의 출력을 받아 실제 피삭체의 가공상태를 다양한 기능을 이용하여 화면상으로 디버깅해 주는

시스템을 구성하였다. 실제로 NC 명령(Command) 내의 오류는 많은 원인을 찾아낼 수 있겠지만, 이는 실제 가공을 행하기 이전에 그래픽 검증 기고(Graphic Proofing Technique)에 의하여 찾아낼 수 있다. 본 논문에서는 CRT 화면상으로, 설계한 그래픽 디버거를 이용하여 피삭체의 가공면을 확대, 회전해 가며 가공상태를 보여줌으로써 NC 프로그램상의 오류를 제거할 수 있는 기능을 제공한다.

2. 전체 시스템의 구성

설계한 전체 시스템은 크게 두 부분으로 나눌 수 있다. CNC 프로그래머로부터 입력을 받아 그래픽 디버거와 CNC 제어부로 각기 출력을 내보내주는 한글 자동 프로그래밍 시스템과, 앞의 시스템으로부터 받은 EIA(Electronic Industries Association) 표준코드 시퀀스를 입력으로 하여 칼라 모니터상에 각종 그래픽 검증기능을 발휘하는 그래픽 디버거로 나뉘어진다.

다시 첫번째 시스템은 CNC 프로그래머에게 문법에 맞는 입력을 강요하는 형식의 편집기(Editor)와, 편집된 파트(Part) 프로그램으로부터 실제 가공의 경로정보를 만들어 내는 주프로세서(Main Processor), 그리고 이 결과 생성된 CL(Cutter Location) 화일을 입력으로 하여 NC 테이프를 만들어 내는 포스트 프로세서(Post Processor)의 세 부분으로 나눌 수 있고, 두번째 시스템은 NC 테이프와 내용이 동일한 정보화일을 입력받아 각종 명령을 해독하고, 그 명령에 따라 피삭체가 각어나가는 모습을 화면상에 보여주고, 최종 가공물의 형상을 확대 및 회전하면서 디스플레이 시킨다. 그림 1은 이상에서 설명한 전체시스템의 데이터 흐름도이다.

3. 한글 자동 프로그래밍 시스템의 구성

(1) 시스템의 기능

한글 자동 프로그래밍 시스템은 기능상으로 그림 2의 내용과 같이 구성된다. 즉, 한글로 정의한 4가지 종류의

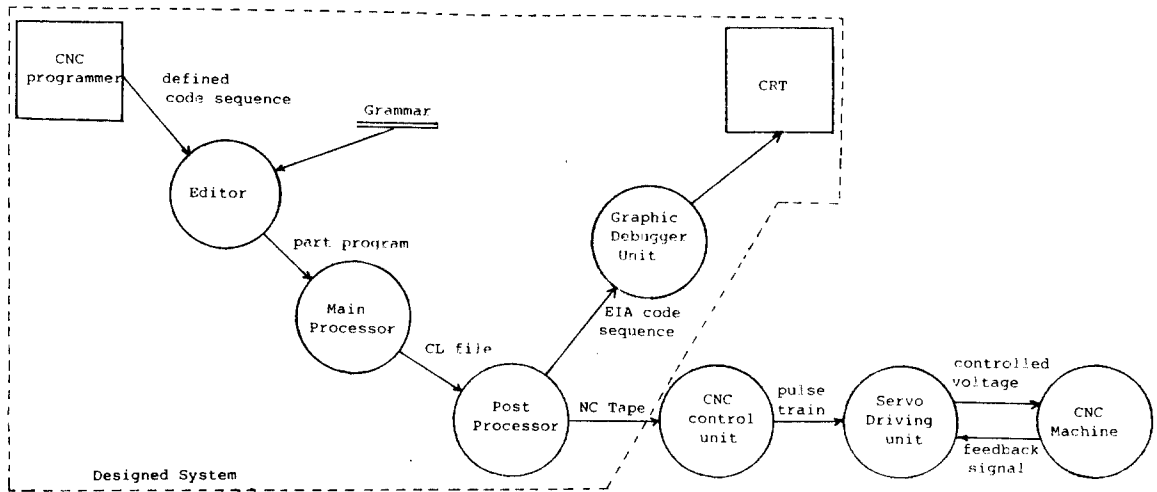


그림 1. 전체 시스템의 데이터 흐름도

Fig. 1. Data Flow Diagram of the Whole System

명령어와, 이를 실행으로 옮겨주는 부분, 그리고 프로그램의 제어, 편집, 상태표시, 자기진단등을 할 수 있는 모니터 명령 ( Monitor Command ), 또 시스템 초기화와 에러처리 부분으로 나눌 수 있다.

(2) 한글 명령어의 정의

1) 운동명령어

- 부터 / 시작점 ; Starting Point Definition
- 가라 / 까지 ; Go/To
- 우로가라 / ; Go Right
- 좌로가라 / ; Go Left
- 앞으로가라 / ; Go Forward
- 뒤로가라 / ; Go Back
- 지나쳐가라 / ; Go Past
- 위로가라 / ; Go On
- 까지가라 / ; Go To

2) 기하명령어

- 점  $n$  = 점 / ; Point Definition
- 원  $n$  = 원 / ; Circle Definition
- 선  $n$  = 선 / ; Line Definition

3) 포스트 프로세서 명령어

- 냉각수 / ; Coolant On/Off
- 운전속도 / ; Feedrate
- 정지 ; Stop(for Tool Change)
- 끝 ; END

4) 보조명령어

- 프로그램 번호 ; PARTNO No  $n$
- 공구지름 / ; Cutter Diameter
- 내부공차 / ; Inner Tolerance
- 외부공차 / ; Outer Tolerance

종료 ; Finish

5) 모니터명령

- 편집 ( 삭제, 삽입, 표현, 탈출 )
- 해독, 실행, 위험정지, 진단, 도움

이상의 각 정의중 '/' 뒤에는 각 명령어 마다의 연산자 ( Operand )가 필요하며, 가능한 한 범위를 축소하여 정의하였다.

(3) 편집기의 설계 및 구성

설계한 편집기는 라인 버퍼 ( Line Buffer ) 형식을 취하지 않고, IBM-PC의 키보드 내의 10개의 기능키 ( Function Key )와 필요한 영자, 숫자, 부호등에 의하여 한자 한자 파싱 ( Parsing ) 해 나가는 방식을 취한다. 여기서 기능키를 이용한 이유는 한글을 사용하기 때문이며, 10개의 기능키로 많은 명령을 처리할 수 없기때문에 전체 명령어 집합 ( Set )을 나무 ( Tree ) 형식으로 분류하여 계층적인 구조로 설계하였다.

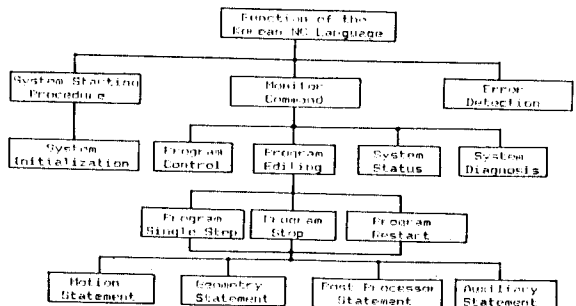


그림 2. 한글 NC 언어의 기능

Fig.2. Function of the Korean NC Language

다음은 각 명령어 집합의 분류이다.

- 제 1단계 ; 편집, 실행, 위험정지, 자기진단 도움 (최상의 단계)
- 제 2단계 ; 삽입, 삭제, 표현, 탈출 (편집단계) 해독, 실행, 단일스텝 실행, 정지, 속행 (실행단계)
- 제 3단계 ; 운동, 기하, 포스트 프로세서, 보조 (삽입단계)
- 제 4단계 ; 앞에서 정의된 22개 명령어, 영차, 숫자, (최하의단계) 부호등

이상에서 상위단계로의 복귀시에는 'F10' 키를 이용하면 된다.

한편, 메모리상에 저장되는 프로그램 화일 (Program File) 에는 정의된 OP - 코드와 ASCII 코드가 저장되며 실행시에 필요한 정보로 바꾸어서 사용하게 된다.

#### (4) 프로세서의 설계 및 구성

프로세서는 크게 두개의 모듈로 분류할 수 있다. 첫째는 주프로세서로서, 이미 정의된 언어에 의해 작성된 파트 프로그램을 해독하여 공구의 경로를 계산해 내고, 각 보조명령어들을 포함하는 CL 화일을 작성하게 된다. 이때 CL 화일에 저장되는 각 정보의 구조는 기존 언어인 APT-III를 참조하여 새로이 정의하였다. 둘째는 포스트 프로세서로서, 주프로세서에서 작성한 CL 화일을 입력으로 하여 특정한 동작기계에 적합한 NC 테이프를 만들어내는 과정을 수행하게 된다. 또한 포스트 프로세서는 그 기능에 따라 다시 입력모듈, 제어모듈, 보조모듈, 운동모듈, 출력모듈로 구분할 수 있다.

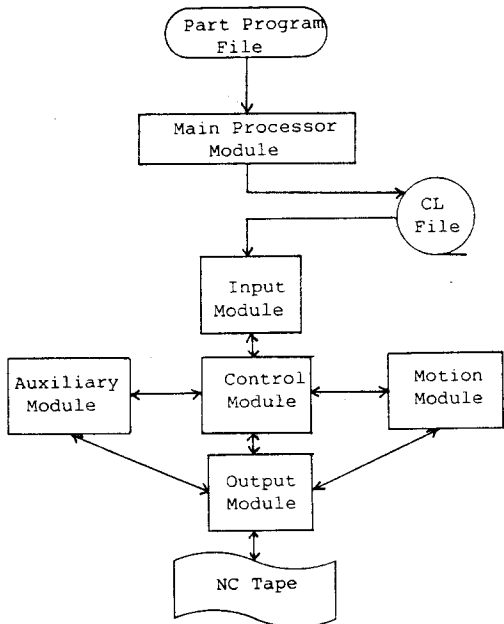


그림 3. 프로세서의 블록선도

Fig.3. Block Diagram of the Processor

그림3은 파트 프로그램을 입력으로 받았을 때의 프로세서의 각 모듈이 어떻게 구성되어 동작하는가를 보여준다.

#### (5) 특성 및 결론

종래 대형 컴퓨터 시스템에서 동작되던 파트 프로그램의 작성과 프로세서의 작업이 점차 소형화, 단순화되어 가고, 또 NC 동작기계와 사람의 관계가 점점 밀접해 가는 현 추세에 맞추어, 우리가 쉽게 접할 수 있는 IBM-PC를 이용하여 매우 기본적인 기능을 갖춘 CNC 언어시스템을 구성하였다. 이 시스템 구성을 위해서 8086/87 어셈블리어 (Assembly Language)와 C 언어를 사용하였으며, 되도록 각 모듈은 독립적인 성질을 갖도록 설계하였고, 미래의 확장이 용이하도록 설계하였다.

#### 4. CNC 선반용 그래픽 디버거의 구성

##### (1) 기능

그래픽 디버거는 NC 프로그램 (EIA 코드 시퀀스) [7] 과 사용자가 최종적으로 원하는 피삭체의 모양을 데 이터 화일의 형태로 입력하여, NC 프로그램의 결과와 최종 피삭체의 모습이 일치 하는가를 검증한다. 이 최종 검증 기능 이외에도 NC 프로그램의 각 시퀀스가 제대로 프로그램 되었는지를 알아보기 위해, 시퀀스에 따라 피삭체가 깎여나가는 모습을 연속으로 보여줄 수 있으며 [8], 툴 (Tool) 이동경로의 궤적을 그릴 수도 있다. 사용자가 화면상에 나타나는 피삭체의 모습을 실감있게 느낄 수 있도록, 피삭체의 3차원 디스플레이를 행하고 간단한 수치 기입 기능을 갖는다. 또한 NC 프로그래밍의 실수로 툴의 날이 아닌 부분이 피삭체에 닿을 경우는 그 즉시, 에러 메시지 (Error Message) 를 내보내 프로그램에 오류가 있음을 알려준다. 부수적인 디스플레이 사양으로서 화면 상에 윈도우 (Window) 를 설정해 피삭체의 원하는 부분을 확대해서 디스플레이 한다.

##### (2) 소프트웨어의 구성

###### 1) 피삭체의 기하학적 모델링

CNC 선반의 가공형상은 피삭체의 회전축을 중심으로 회전하면서 깎여나감으로 회전축을 포함하는 임의의 각도의 면을 만들었을 때, 나사 절삭을 제외하고는 그림 4와 같이 모든 단면이 같은 모양을 갖고 있고, 그 단면은 회전축을 중심으로 상하대칭이 된다. 그러므로 회전축을 포함하는 단면의 회전축 윗부분에 관한 2차원적 정보를 가지고 있으면 회전축을 포함하는 단면을 생성해낼 수 있고, 회전축을 포함하는 단면에 관한 정보를 안고 있으면, 피삭체의 3차원적 형상을 생성해낼 수 있게되어 CNC 선반용 피삭체의 기하학적 모델링은 2차원 평면상의 다각형을 모델링하는 문제로 귀착이 된다. 본 시스템에서는 다각형의 꼭지점 (Vertex) 의 좌표와 각 꼭지점 간의 선의종류 (직선 또는 원호) 를 기억함으로써 2차원 다각형의 정보를 가능한 정밀도까지 저장한다.

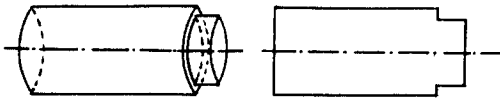


그림 4. 일반적인 피삭체의 모습

Fig.4. General View of Lathed Object

## 2) 프로그램의 구성

### 가. 최종가공형상 및 프로그램 시퀀스 입력

최종가공물의 형상과 NC 프로그램에 정의된 데이터 파일의 형식으로 입력 받는다. 이때 입력된 최종가공물의 형상은 NC 프로그램의 결과와 비교되어, NC 프로그램이 원하는 최종가공물을 생성할 수 있는지의 여부를 판단하는데 쓰여진다. NC 프로그램은 디스켓에 저장된 텍스트 파일 (Text File)의 형태로써, IBM-PC에서 수행되는 에디터 (Editor)를 사용하여 작성을 한다. 에디터 종류에 제한은 없으며, 파일의 맨 끝에 파일의 끝 (End of File)을 알리는 ^Z (Control-Z)를 삽입하여야 한다.

### 나. 디스플레이 사양 및 윈도우 설정

사용자와 대화식으로 설정한 사양에 따라서 화면 윈도우를 설정하므로써 피삭체의 부분을 확대해서 디스플레이 시킬 수 있고, 피삭체와 툴을 동시에 그려 나가면서 피삭체가 깎여나가는 모습을 애니메이션 (Animation) 할 것인지, 툴경로의 궤적만을 그릴 것인지 결정한다.

### 다. 형상데이터 변환 및 동작 디스플레이

프로그램을 한줄씩 읽어들이면서 툴의 움직임으로 인해 생기는 피삭체의 절삭상태를 연산해내어 피삭체의 형상정보를 변화시키고, 변화된 모습을 디스플레이

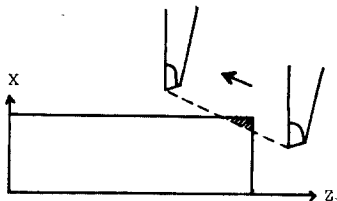


그림 5. 툴의 움직임으로 인한 피삭체의 변화

Fig.5. Effect of Tool Motion on the Object

한다. 툴의 움직임으로 인하여 툴의 궤적이 바뀌는 다각형을 생성해내고 피삭체와 겹쳐지는 부분 (그림 5에서 빗금친 부분)이 피삭체에 포함이 안 되도록 피삭체의 형상정보를 재조작 함으로써 형상정보의 최신화 (Update)가 이루어진다. 이 과정에서 툴의 날이 아닌 부분이 피삭체에 접촉이 되면 이것을 감지하여 에러 메시지를 발생시킨다.

### 라. 수치기입 및 3차원 디스플레이

사용자의 요구에 따라 피삭체를 모델링한 다각형의 각 꼭지점 (Vertex)의 (x, z) 좌표를 디스플레이 시키거나 2차원 피삭체 정보를 이용하여, 3차원 형상을 생성해낸다.

### (3) 프로그램의 특성 및 결론

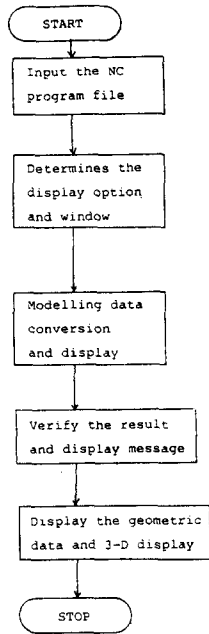
현재 이 시스템은 CNC 선반의 여러동작 기능중 기본적인 두가지 (직선보간, 원호보간)를 대상으로 하였고, 툴의 선택도 사용자가 원하는 툴의 종류를 프로그램내에 등록 시키기가 다소 복잡한 단점이 있다. 피삭체의 기하학적 모델링과 툴의 움직임으로 인한 피삭체 형상정보의 변환 방법등 그래픽 디버거에 필요한 기본적인 알고리즘이 완성되었으므로 직선과 원호를 제외한 보간방법과 나선절삭, 반복동작등의 여러기능을 처리가능하게끔 프로그램 확장을 할 수 있고, 일반적인 툴의 등록을 위한 프로그램도 개발중이다. 이러한 프로그램 확장을 고려하여 프로그램 구성에 있어서 모듈래티 (Modularity)를 강조하였고, 프로그래밍 언어도 C 언어를 사용하였다. 이 시스템의 입력으로는 일반적인 NC 프로그램이 사용되므로 자동프로그래밍 시스템이 없어도, 단독으로 운용이 가능하다. 피삭체 기하학적 모델링 방법의 단순화 및 이에 알맞는 피삭체 형상 정보 변환 알고리즘의 개발로 인하여 기존의 IBM-PC에 별도의 메모리를 확장할 필요 없이, 피삭체의 기하학적 데이터를 원하는 수준의 정밀도까지 저장하는 그래픽 디버거를 구성할 수 있었다.

### 5. 결론

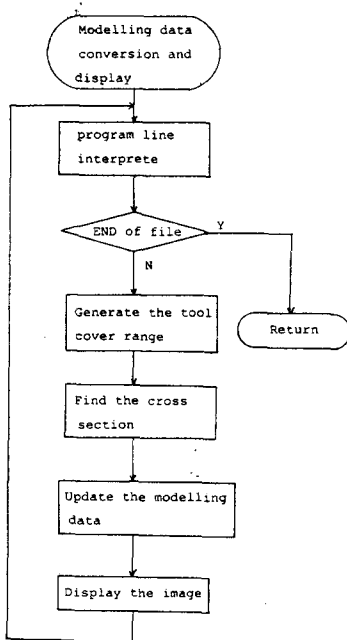
시스템에서 실제 공작물을 깎기전에 그래픽으로 잘못된 부분을 찾아 프로그램상의 오류를 제거함으로써 사용자에게 편의를 제공할 수 있는 그래픽 디버거를 설계하였다. 또한 새로이 정의한 한글 자동프로그래밍 언어를 운영하는 시스템을 구성하고, 이로부터 얻은 데이터 파일을 이용하여, 실제 피삭체의 가공상태를 IBM-PC 화면상에 보여 줄 수 있었다. 앞으로 절삭이론을 고려하고 일반적인 툴의 등록을 위한 프로그램의 개발과 범용 NC 공작기계에 적용할 수 있는 일반화된 한글 자동프로그래밍 언어의 개발이 이루어져야 하겠다.

참고 문헌

- [1] 이봉진, NC 강의, P84, 성인당, 1986.
- [2] Roger S. Pressman and John E. Williams, Numerical Control & Computer Aided Manufacturing, P176, Wiley, 1977.
- [3] Robert Jourdain, Programmer's Problem Solver for the IBM PC, XT & AT, Brady C.C., 1985.
- [4] Robert Lafore, Assembly Language Primer for the IBM-PC & XT, The Waite Group, 1984.
- [5] Peter Abel, Assembler for the IBM-PC and PC-XT, Reston Pub. Com., 1984.
- [6] FANUC Ltd, FANUC SYSTEM 6T-MODELB Operator's Manual, 1983
- [7] 윤종학, NC 프로그래밍, 연화, 1985.
- [8] Mikell P. Groover, and Emory W. Zimmers, Jr. CAD/CAM, pp.146-184, Prentice Hall, 1984.



- ㄱ. 주프로그램
- ㄴ. Main Routine



- ㄴ. 데이터 변환 루틴
- ㄴ. Data Conversion Routine

그림 6. 그래픽 디버거의 흐름선도

Fig.6. Flowchart of Graphic Debugger