

## 선반용 대화형 프로그래밍 시스템(I)

신 동 수\*, 정 성 중\*\*

### Conversational Programming System for NC Lathes (I)

Dong-Soo Shin\*, Sung-Chong Chung\*\*

#### Abstract

A conversational programming system for turning processes was studied to enhance the user friendliness of a NC Lathe by adopting man-machine interface functions through Delphi programming tool under the Windows 95 environment. Functions of tool collision checking were developed through zone limitation algorithms. Final shape of workpiece was generated by means of tree structure algorithm. Shop floor programming performance was incorporated in the developed CAM module. In order to increase flexibility of the man-machine interface, graphic based programming tools have been developed. A NC turning machine equipped with an open architecture PCNC was used as a test bed of the system. Performance of the system was verified through case studies.

Keywords : Conversational Programming (대화형 프로그래밍), Graphic-Based Programming (그래픽기반형 프로그래밍), Machining Process (가공공정), SFP (Shop Floor Programming : 작업장프로그래밍), Tool Database(공구데이터베이스).

#### 1. 서 론

요사이 생산현장에서는 다품종소량생산체제와 신속한 제품개발을 위해 대화형 프로그램 (Conversational Program) 에 많은 관심을 가지게 되었으며, 숙련작업자의 감소와 경험이 적은 작업자가 간단하게 작업할 수

있는 환경요구 등 대화형 프로그램에 대한 관심도가 높아지고 있다.<sup>(1-3)</sup>

수치제어 공작기계의 작업명령 언어인 파트 프로그램을 작성하기 위해 CAM 시스템을 이용하고 있으나, 프로그램을 습득하는데 많은 시간이 요구되며, 가공현장과 독립된 프로그래밍 공간은 물론 컴퓨터가 요구되며 밀링에

\* 화천기공(주), 기술개발 연구소

\*\* 한양대, 기계공학부

비해 단순한 선삭 가공을 위한 CAM 시스템의 채용은 경제성이 떨어지는 편이다.<sup>(4)</sup> 이 때문에 CNC 시스템 제조사에서는 자사의 수치제어 장치에서만 사용 가능한 프로그래밍 도구를 개발하여 제공하고 있다. Siemens 사의 Blue-print / Support 프로그래밍과 WOP 시스템, MAZAK 사의 대화형 프로그램, MORISEIKI 사의 MORIC-3Y, Enshu 의 TOMATO<sup>(5)</sup>, YASNAC 사의 Compact 프로그래밍, OKUMA 사의 Raku Raku 프로그래밍, FANUC 사의 Super-Cap, Symbolic-Cap<sup>(6)</sup> 등이 그 예이다.

그러나, 요사이 개방구조 (Open Architecture) NC 체제의 요구와 MMI (Man-Machine Interface) 의 개방요구때문에 기존의 전용 OS 환경이 아닌 Windows 환경 하에서의 PCNC 개발이 이루어지고 있으며, 가공능력이 강화된 프로그램이 필요하게 되었다.<sup>(7)</sup>

본 논문에서는 개방구조 NC 에 적용할 수 있는 Windows 환경하의 선반용 다기능 대화형 프로그램 개발에 대하여 연구하였다. 다목적 도형데이터 정의, 가공공정을 위한 공구데이터 정의, 그리고 공구와 공작물의 간섭, 충돌회피를 위한 시험가공 알고리즘 개발을 통해 가공프로그램의 단시간 편집 및 작성기법을 제안하였다. 아울러 그래픽 기반형 프로그래밍 기법을 사용함으로써 정확하고 신속한 프로그래밍 방법론을 제시하였고, 작업장프로그래밍 (Shop Floor Programming: SFP) 과 CAM 시스템의 혼합을 통해 보다 친숙한 대화형 프로그램 환경을 제안하였다.

## 2. 가공프로그래밍 시스템의 분류

가공프로그램은 온라인 (On-Line) 혹은 오프라인 (Off-Line) 으로 생성할 수 있다. 온라인의 의미는 공작기계 제어기상에서 직접적으로 프로그래밍할 수 있는 것을 말하며, 오프라인의 의미는 공작기계가 아닌 다른 곳에서 파트프로그램을 생성하는 것을 의미한다. 각각의 방법은 장점과 단점을 가지고 있으며, 어디에서 프로그래밍을 할 것인가 보다는 어떻게 프로그램을 생성할 것인가가 더 중요한 문제로 대두된다. 가공프로그래밍은 다음과 같이 수동프로그래밍 (Manual Programming), 작업장프로그래밍 및 CAM 으로 나눌 수 있다.

### 2.1 수동프로그래밍

Fig. 1 은 수동 프로그래밍에 대한 작업흐름도이다. 라인 대 라인 (Line by Line) 을 통해 프로그램을 작성하게 되고 각 제조사마다 다양한 가공코드 형식, 수치계산 및 기하학적인 지식이 요구되며, 프로그래머는 절삭에 따른 부품상의 특별한 사양을 계산하고 편집기 상에서 프로그램을 작성해야 한다.

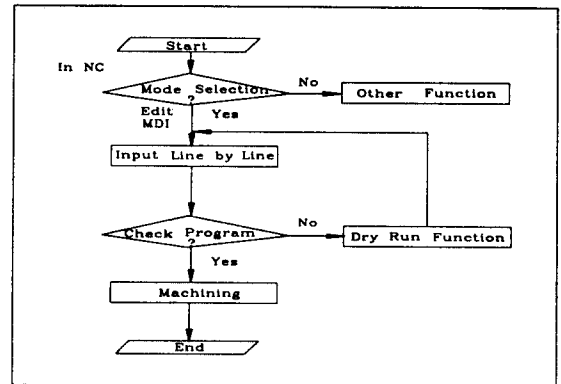


Fig. 1 Flow chart for Manual Programming

그러나, 이 경우 복잡한 부품은 프로그램하기가 어렵거나 불가능하게 된다. 수치계산, 프로그램 편집 및 입력에 대한 오류가 빈번히 발생하며, 수동 프로그래밍 기법으로 작성된 프로그램의 검증은 위해서는 드라이 런 (Dry Run) 의 기능이 필요하게 된다.

### 2.2 작업장프로그래밍

Fig. 2는 작업장프로그래밍의 경우 작업흐름도이다. 제 1 세대 SFP 제어는 빈칸을 메우는 (Fill in the Blank) 접근법을 통해 프로그래밍 하는 대화형이었으나 최근에는 대화형 그래픽 SFP 제어 가 등장하였으며, 여기서는 부품의 기하모형 생성, 공구제작 생성 및 검증을 문자가 아닌 그래픽을 통해 실행할 수 있게 되었다.

또한, SFP 제어는 프로그램 상에서 수치계산을 수행할 수 있고, 가공 중에 프로그램을 작성할 수 있는 백그라운드 프로그래밍 (Background Programming) 기능을 제공하고 있다.<sup>(7)</sup> 이 때문에 SFP 는 협소한 작업장에서 동일한 작업자가 부품을 프로그래밍하거나 작업을 셋업할 경우에 유리하고, 작업의 효율성 판단을 통해 직접적으로 프로그래밍 하거나 대화로써 작업을 수행하는 것을 선택

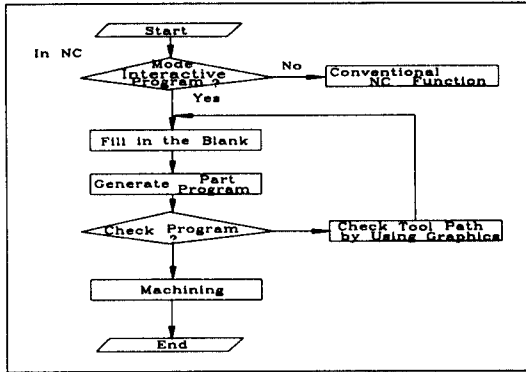


Fig. 2 Flow chart for Shop Floor Programming

할 수 있다. 그러나, 열악한 작업환경에서는 편리하게 작업 할 수 없으며, 복잡한 형상을 작업하는 경우에 어느 정도 제약을 지니고 있다. 그러므로 상대적으로 단순한 부품 가공에 사용되며 일 대 일로써 작업을 수행하기 때문에 다른 공작기계에서 어떻게 프로그래밍할 것인가의 문제를 해결할 수 없게 된다.

### 2.3 CAM 시스템

Fig. 3 은 CAM 작업흐름도이다.<sup>(8)</sup> CAM 시스템은 부품의 형상을 정의하는 모델러 기능, 공구선택과 공구궤적을 생성하고 기계가 이해하는 형식으로 변환하는 포스트프로세서 (Postprocessor) 기능을 가지고 있다.

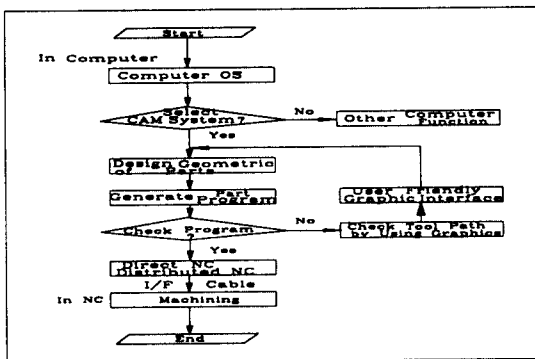


Fig. 3 Flow chart for CAM system

Fig. 4 는 숙련도, 제어축수 및 대화의 요구도에 따른 가공시스템의 분류를 보인 것이다. 기초 프로그래머와 작업자의 경우 대화의 요구도는 높으며 숙련가일수록 대화의 요구도는 낮아진다. 따라서 대화성이 높고 다축을 수용할 수 있는 초보자용의 프로그래밍 환경이 가공프로그래밍 시스템에서 요구된다.

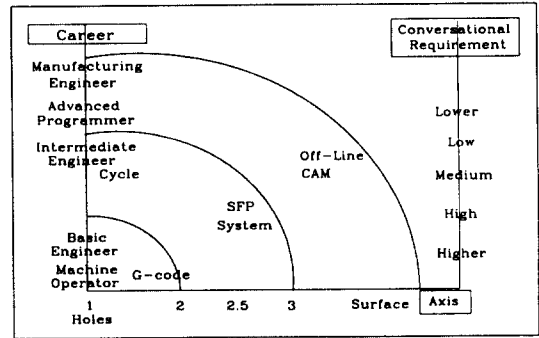


Fig. 4 Classification of various programming systems

### 3. 개발환경 및 구성모듈 설계

전절에서 소개한 기존의 SFP 시스템과 CAM 시스템의 문제점을 해결하기 위해서는 SFP 시스템과 CAM 시스템의 혼합을 통한 대화형 프로그래밍 시스템이 요구된다. 개방형구조 (Open Architecture) 에 대응하는 시스템 구성을 위해 Windows 95 환경을 선택하였으며, Fig.

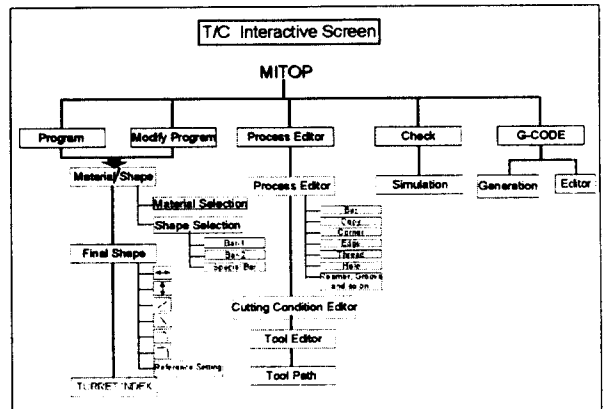


Fig. 5 Structure of the Developed System

5는 Borland 사에서 제공하는 Delphi 프로그램 Tool<sup>(9)</sup>을 이용하여 개발된 프로그래밍 시스템의 구조이다. 그림과 같이 형식상으로는 Program, Modify Program, Process Editor, Check, G-code 등 5 가지 부시스템으로 구성되어 있으며, Program 메뉴 선택시 가공프로그램의 신규작성은 다음과 같은 6 가지 모듈에서 수행될 수 있도록 설계하였다.

1. Material / Shape 모듈 : Table 1 은 개발된 시스템에서 공작물과 공구 재질에 따라 절삭조건을 선정하기 위해 필요한 구조체 데이터와 가공조건 변수를 보인 것이다.

Table 1 Definition of Structure of Variables for Workpieces and Tools

Data Type	Variable	Contents			
typedef struct! ..... ! CUTCONDITON:	UINT m_WpMaterial	Workpiece Material Code 0 : FC25    1 : FC45 2 : SS41    3 : S45C 4 : SKD11   5 : SUS304 6 : AL      7 : BRASS			
		UINT m_OperationType	Type of Machining 0 : ROUGHNESS   1 : MIDDLE 2 : FINESS       3 : THREADING 4 : GROOVING    5 : CENTER DRILL 6 : DRILL		
			UINT m_ToolMaterial	Tool Material 0 : HSS 1 : CBN 2 : CMNG 3 : CERMET	
				DOUBLE m_CutDepthMin	Depth of Cut (Minimum) mm
				DOUBLE m_CutDepthMax	Depth of Cut (Maximum) mm
		DOUBLE m_CutSpeedMin	Spindle Speed (Minimum) m/min		
		DOUBLE m_CutSpeedMax	Spindle Speed (Maximum) m/min		
	DOUBLE m_CutFeedMin	Cutting Feedrate (Minimum) mm/Rev			
	DOUBLE m_CutFeedMax	Cutting Feedrate (Maximum) mm/Rev			

2. Final Shape 모듈 : Fig. 5 의 Final Shape 모듈과 같이 선, 사선, 원, 원호, 라운드, 모따기 등의 요소로 최종형상을 정의한다.
3. Process Editor 모듈 : 다양한 가공동작을 1 개의 블록 (Block) 으로 지령하여 효율적으로 프로그램을 작성할 수 있도록 하는 모듈이다. 봉재가공의 6 가지, 나사가공의 2 가지, 코너가공의 6 가지, 모방가공의 6 가지, 단면가공의 6 가지, 홈가공의 6가지, 구멍가공의 5 가지등 총 37 가지 사이클이 존재한다.

4. Tool Editor 모듈 : Fig. 5 의 Tool Editor 에서는 공구재질, 공구형상, 터릿번호, 공구치수 및 공구 종류 등을 선택하여 Table 2 와 같은 구조체 형태로 가공에 쓰일 공구번호, 형태, 길이 및 공구편 집을 위한 데이터를 정의한다. 공구 데이터는 절삭 조건과 관련되어 데이터베이스에 서 관리한다.

Table 2 Definition of Structure of Variables for Tools

Data Type	Variable	Contents	
typedef struct! ..... ! TOOL:	UINT m_ToolID	Tool Number	
	UINT m_ToolType	1. OUTER SURFACE   2. INNER SURFACE 3. END FACE 4. EXTERNAL/INTERNAL THREAD 5. GROOVING   6. DRILL   7. TAPPER 8. CENTER DRILL 9. END MILL   10. SIDE CUTTER   11. CHAMFER 12. REAMER	
		UINT m_Direction	TOOL Direction : 1. LEFT   2. RIGHT
		UINT m_MillTurn	Machining Point of DRILL . CENTER DRILL . TAPPER
		UINT m_RoughFine	Accuracy of Machining
		REAL m_SpindleDirection	Rotation Direction of SPINDLE
		REAL m_ToolLength	tool + holder
		UINT m_Output	Output T Code through Part Program
		REAL m_NoseAngle	Angle of TIP
		REAL m_NoseRadius	Radius of TIP
		REAL m_NoseWidth	Width of TIP
		REAL m_VirtualTipPosition	
		REAL m_CutLength	Length of Real Machining
	REAL m_Pitch	Pitch of TAPPER	
	REAL m_ToolRadius	1. Drill   2. Tapper   3. Center Drill 4. Sider Cutter 5. Chamfer   6. Reamer	
	REAL m_ToolSmallDia	Chamfer	
	REAL m_ToolNotUse	Chamfer	
UINT m_BladeNumber	Sider Cutter		
UINT m_ToolMaterial	Material of TIP		

5. Cutting Condition Editor 모듈 : Fig. 5 의 Cutting Condition Editor 에서는 절삭계수인 이송속도, 스펀들속도, 절삭깊이 및 표면조도 등을 정의한다.
6. Postprocessor 모듈 : Fig. 5 중의 G-code 와 Check 부분은 현 가공물의 확대 및 축소 등과 같이 보는 각도를 자유자재로 할 수 있도록 하는 기능을 수행하는 시뮬레이션 부분과 가공코드 생성 및

편집 모듈을 가지고 있어 제어기 종류에 따른 가공 프로그램을 생성한다.

### 4. 프로그램 구조 및 기능

#### 4.1 도형정의 알고리즘

Fig. 6 은 가공물의 최종형상을 Table 3 의 도형정의 구조체 변수를 이용하여 선, 사선, 라운드, 모따기 등의 입력값에 따라 생성한 경우를 보인 것이다. 먼저, 출발 원

Table 3 Definition of Structure of Variables for Geometric Elements

Data Type	Variable	Contents
typedef struct {	UINT m_Model ..... } GEO;	0 : Vertical Line 1 : Horizontal Line 2 : Skew Line 3 : Arc 4 : Chamfer 5 : Round
UINT m_Tree		0 : Left 1 : Right
REAL m_Coordinate		0 : Start / End Point 1 : Radius 2 : Angle 3 : Length
UINT m_Display		0 : On 1 : Off
UINT m_Number		Geometric Element Number
UINT m_Dummy	unused variable	

점인 x 와 z 좌표값을 정의하며, 정의가 없는 경우는 기본값으로 그림과 같은 위치에 원점이 정의된다. 도형정의는 상대 혹은 절대좌표값으로 정의되며, 입력의 정확성을 위해 현재위치를 화면상에 표시해 준다.

Fig. 7 왼쪽부분의 Tree 구조<sup>(10)</sup>는 모델의 형태 파악이 용이하도록 프로그래밍된 도형모델구조 형태를 보여주며, 원점을 x 축 상에 설정한 경우이다. Table 3 과 같이 UINT m\_Tree 에서 현재방향과 같은 방향의 가지는 동일한 형태의 도형정의를 의미하며 현재방향과 반대방향의 가지는 다른 형태의 도형정의를 나타낸다. 또한, 도형정의의 수정 편집을 위한 이전 도형과 현재 도형과의 관계를

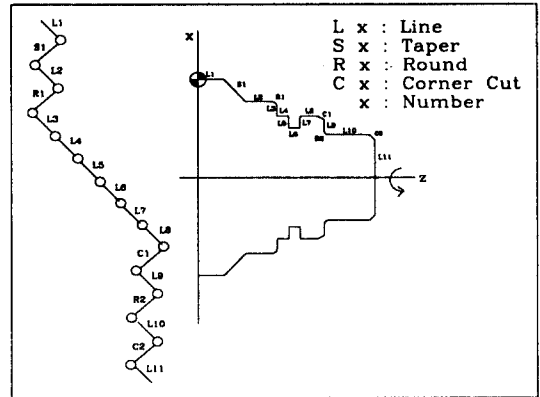


Fig. 7 Model Shapes with Tree Structure (Start Point = on X axis)

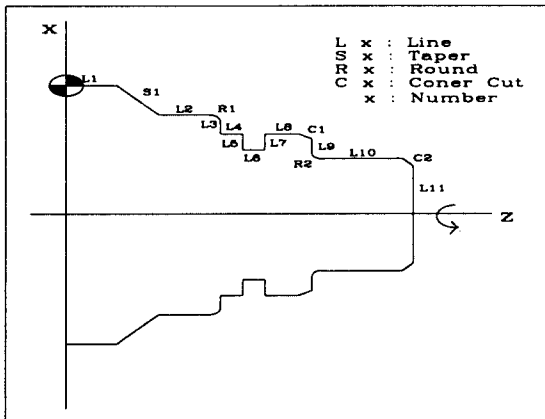


Fig. 6 Part geometry

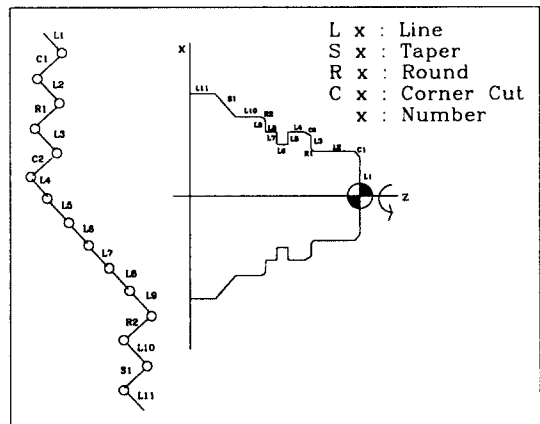


Fig. 8 Model Shapes with Tree Structure (Start Point = on Z axis)

표현하기에 편리하다. Fig. 8 은 원점을 z 축에 설정한 경우를 보여준다. Fig. 7 과 비교하여 처음과 끝부분 Tree 구조의 도형모델구조 형태가 교체되었음을 알 수 있다. 이러한 Tree 구조는 모델요소의 차별화와 관계화에 편리성을 가지고 있으므로 공정입력에 용이한 데이터베이스를 형성시킬 수 있다.

Fig. 9 는 가공물의 윤곽을 정의하기 위한 흐름도를 나타내며, 윤곽정의 (최종형상) 는 Table 3 의 UINT m\_Model 중 도형의 기본요소인 선, 사선, 라운드, 모따기 등의 좌표값을 통해 정의된다. 그림과 같이 먼저 가공물의 원점을 설정하고 설정된 원점으로부터 각 도형요소의 좌표값을 정의한다.

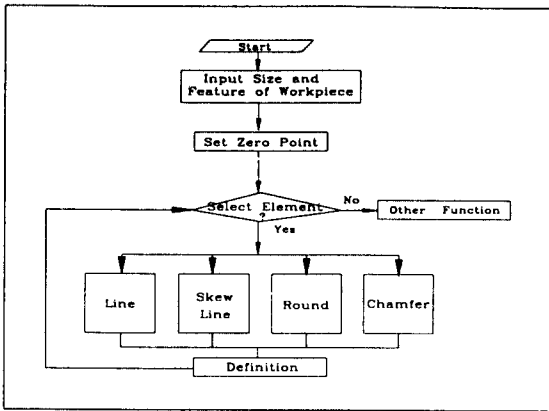


Fig. 9 Flow chart for Contour Definition

#### 4.2 공정계획

공정계획<sup>(11)</sup>은 취급 정보량이 많고 그의 결정기구에 경험적인 부분이 포함되는 등의 개발상 문제를 내포하고 있다. 여기서는 기계가공에 한정하여 공정입력의 역할과 처리에 대한 사항을 제시한다.

Fig. 10 은 공정계획에서의 취급정보를 보여주며 Table 4 는 공정관련 데이터 구조를 보여주고 공정형태에 따른 공구보정, 공차 및 가공조건과 공구의 접근점과 도피점 등을 설정할 수 있다. 여기서, Fig. 10 의 세부적인 항목을 살펴보면 다음과 같다.

1. Part Information 모듈 : 부품의 완성형상을 표시하는 기하형상정보를 가지며, 기술정보는 마무리정도, 표면거칠기 및 치수 등을 표시한다. 재질 및 소재의 처리와 같은 소재의 가공법을 표시하는 소재정보는 환봉, 육각봉 및 그외의 형태의 봉을 선택할 경우 취급되며, 조립한 상태를 추정할 수 있는 부품 접속정보는 표면거칠기와 형상정보에 의해 취급 된다. 그리고 납기, 생산요구량, 생산량 등의 정보는 메시지 형태로 취급되어진다.

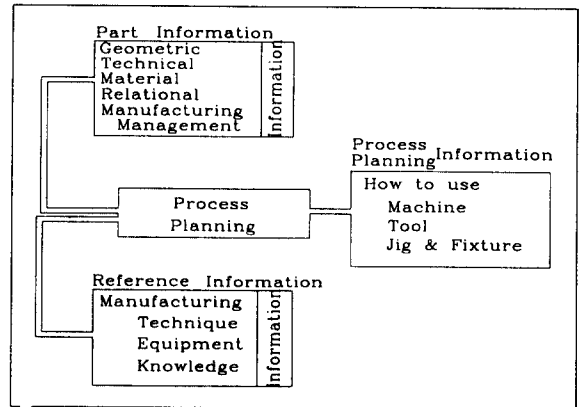


Fig. 10 Information of Process Planning

2. Reference Information 모듈 : 홈, 구멍과 같은 가공형상특징과 가공방법과의 대응관계 그리고 가공 조건과 공구의 관계 등을 포함하는 터릿 윈도우를 통해 생산기술정보를 가지며, 소유하고 있는 공작기계, 설비 공구 등의 능력 및 제원에 관계된 공구편집과정의 생산설비정보가 있다. 그리고 설비 등을 효율적으로 사용하기 위한 노하우와 부품 및 제품 정도향상을 위한 노하우 등의 생산지식정보가 있다.
3. Process Planning Information 모듈 : 적절한 공작기계와 그의 가공순서, 적절한 공구와 그의 사용 순서 그리고 적절한 치공구의 사용 순서가 공정편집 및 가공조건 입력 등에 위치한다.

Table 4 Definition of Structure of Variables for Processes

Data type	Variable	Contents
typedef struct	UINT m_OperationType	Variables of Machining Type 0 : Bar 1 : Pattern Repeating 2 : Residual 3 : End Facing 4 : Threading 5 : Grooving 6 : Center Drill 7 : Drilling 8 : Tapping
OPERATION:	UINT m_ShapeID(MAX SHAPE)	ID of Machining Shape
Typedef struct	UINT m_ToolNum	Tool Number
PCONDITION:	UINT m_RadiusComp	Tool Compensation 0 : Don't use Tool Compensation ( G40 ) 1 : Compensate Right ( G41 ) 2 : Compensate Left ( G42 )
	REAL m_Allowance_x	Machining Allowance of Finishing in X 0.0 ~ Maxium Allowance
	REAL m_Allowance_z	Machining Allowance of Finishing in Z 0.0 ~ Maxium Allowance
	REAL m_CutDepth	Depth of Cut in First time 0.0 ~ Maxium Cutting Depth
	UINT m_Roughness	Surface Roughness 0.0 ~ Maxium Roughness
	UINT m_SpindleSpeed	Spindle Speed
	UINT m_fFeedTye	Method of Cutting Feedrate 0 : mm/Min ( G98 ) 1 : mm/Rev ( G99 )
	DOUBLE m_FeedRate	Feedrate
	BOOL m_fCoolant	Coolant 0 : On 1 : Off
	UINT m_fSpindleStop	Spindle Rotation after Machining 0 : Stop 1 : Rotation
	DOUBLE m_Appr[2]	Approach Point to Workpiece X, Z
	DOUBLE m_Esc[2]	Escape Point after Process X, Z

Fig. 11은 정보처리 순서를 보인 것이며 세부적인 내용을 보면 다음과 같다.

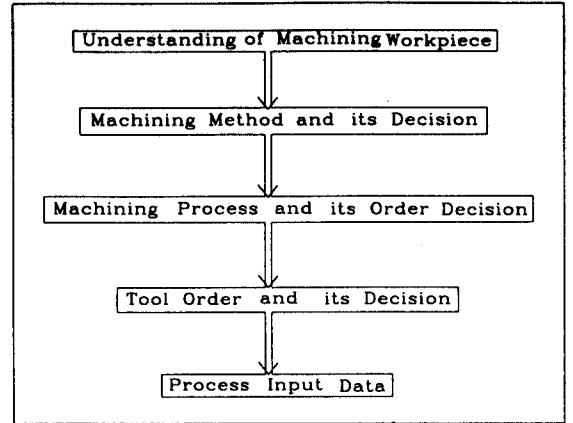


Fig. 11 Sequence of Data Processing

1. 가공대상과 그의 이해 : 도면을 통한 모델러를 이용한 부품구성표가 기재되어 있는 정보와 Table 3의 도형정의와 구성 및 다른 도형과의 접속관계의 정보를 이해하고 형상특징을 인식하는 것이다.<sup>(11,12)</sup>
2. 가공방법과 그의 순서결정 : Fig. 12는 황삭가공의 방법에 대한 결정을 위해 형상을 분할한 예이다.<sup>(13)</sup> 공구의 조건과 절삭조건 및 공차 등의 도면상황의 고려를 통한 사용자의 결정이며, (a) 경우보다는 (b)와 (c)가 정도 및 시간적인 측면에서 효율적이다.

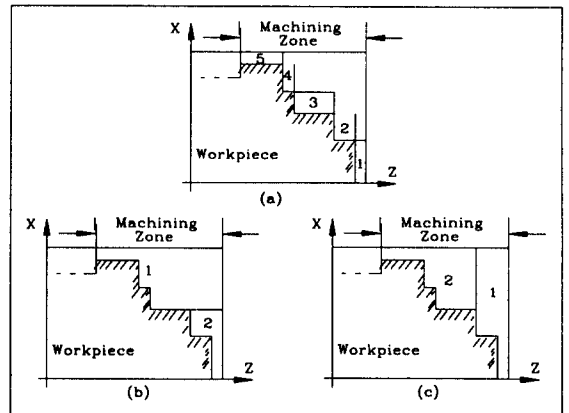


Fig.12 Machining Methods for Rough Machining

3. 공구후보의 결정 : 가공면 및 가공형상특징은 절삭 운동과 이송운동에 대하여 공구의 Nose 형상과의 조합을 통해 생성된다. 이에 따른 상세한 절삭조건 결정과 공구의 특정한 작업설계에 대해 사용 가능한 공구의 후보가 선택되어 진다.

### 4.3 사례연구

Fig. 13 은 Borland C++ 의 Delphi 를 이용하여 프로그래밍된 도형정의 즉 최종형상을 구성하기 위한 메뉴를 나타낸 것이다. Table 1 ~ 2 의 공작물과 공구의 재질 등을 선택한 후 Table 3 의 도형정의를 Fig. 13 과 같이 수직선, 수평선, 왼쪽사선, 오른쪽사선, 라운드, 모따기의 6 가지 기본요소로 사용할 수 있도록 구성되어 있다. Fig. 13 의 상단부분은 구성된 기하학의 요소에 대한 확대 및 축소기능을 나타내는 메뉴를 보여주고 있다.

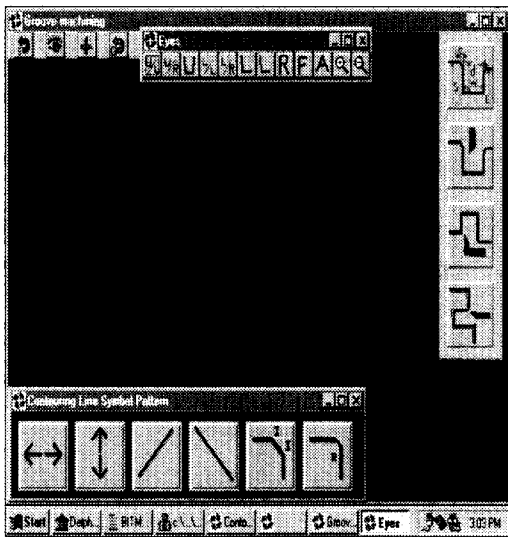


Fig. 13 Menu of Geometric Elements for Final Shape

Fig. 14는 공정입력모델 중 축가공의 경우에 대한 입력모델을 나타낸다. Table 1 과 2 의 내부적인 프로그래밍 구조를 갖는 공구는 일반적인 공구모델중에 하나를 선택하는 과정을 보여주며, 맨 위의 Icon 은 축가공을 위한 입력 파라미터를 나타내며, 내가공과 외가공 및 공구자동교환기능이 있는 내외가공의 기능을 갖추고 있다. 또한,

오른쪽 상단과 같이 모듈을 선택할 경우 위치에 따라 도움말이 디스플레이 됨으로써 사용자의 편의를 도모하였다.

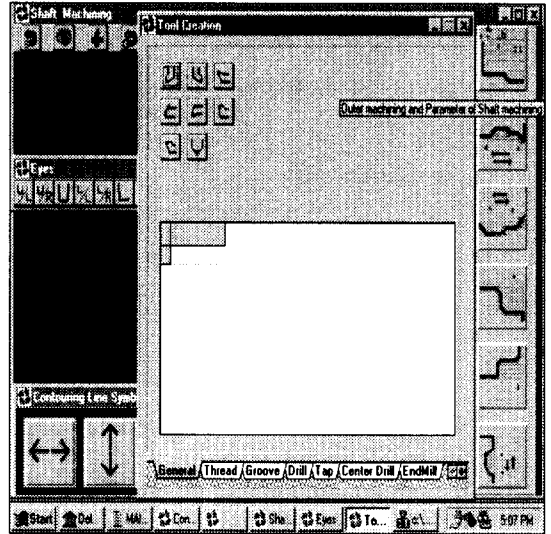


Fig. 14 Menu of Tool Selection and Machining Type for Shaft Machining

Fig. 15는 공정입력모델 중 나사가공의 입력모델과 외경, 내경에 대한 입력모델을 나타내고 있다. 맨 위의 Icon 은 나사가공을 위한 입력 파라미터를 보여주며, 가공부위의 나사 종류와 공구의 선택을 나타낸다.

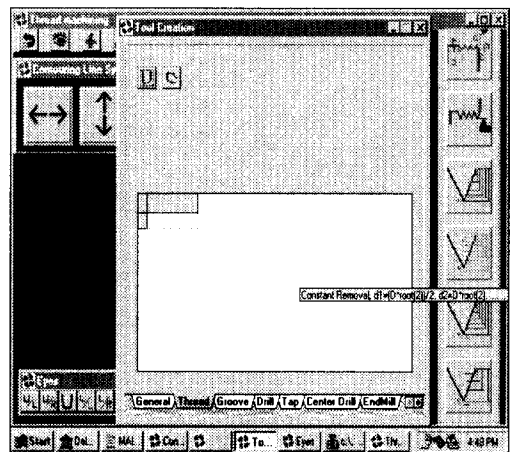


Fig. 15 Menu of Tool Selection and Machining Type for Thread Cutting



Fig. 16 은 구멍가공의 공정입력모듈을 나타내고 있으며, 이것은 심압대 (Tailstock) 혹은 C 축 (터릿에 하나의 제어축수를 이용하여 공구를 회전시키는 축)의 형태를 이용하는 가공으로 센터링, 리머, 보링 및 탭핑 등의 입력모듈을 오른쪽의 Icon 을 통해 실행할 수 있도록 하였다.

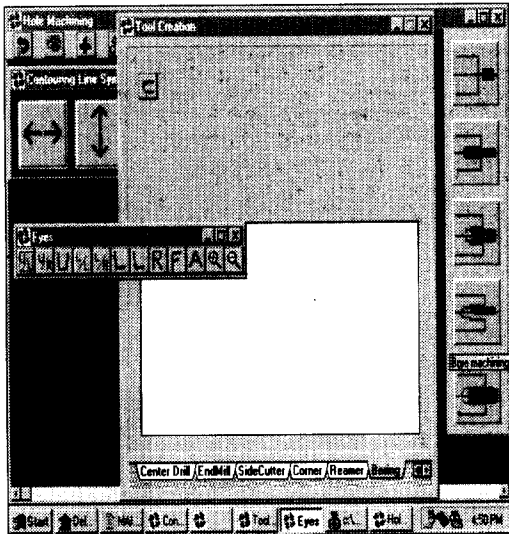


Fig. 16 Menu of Tool Selection and Machining Type for Hole Machining

## 5. 공구계적 및 검증

### 5.1 공구계적

Fig. 17 과 같은 최종 가공물 윤곽형상은 Fig. 9 도형 정의에 대한 흐름도를 이용하여 정의되며, 정의된 가공물은 Fig. 11 의 공정입력을 통해 데이터가 입력된다. 또한, 정확한 입력을 위해 현재 좌표값을 화면에서 볼 수 있다. 최종의 목표인 공구계적을 생성하기 위한 전체적인 알고리즘은 Fig. 18 과 같다. 공작물의 최종형상 정의에서부터 가공영역 선택 및 만족스러운 공구계적까지의 흐름도를 보여준다.

Fig. 19 에서는 공구계적이 만족스럽지 못하면 가공영역의 절삭조건 및 가공영역 자체를 수정할 수 있도록 Fig. 5 메뉴중의 공정편집을 통해 유연성을 부여하였다.

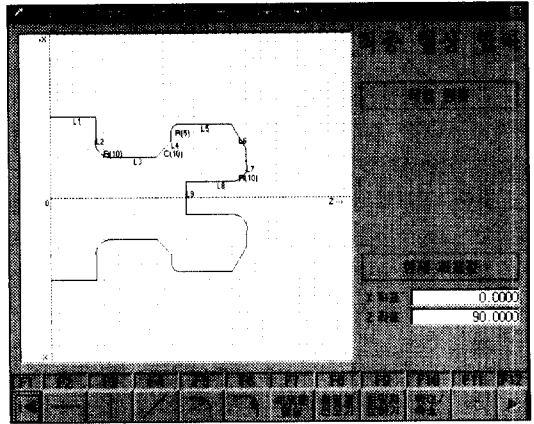


Fig. 17 Final Shape

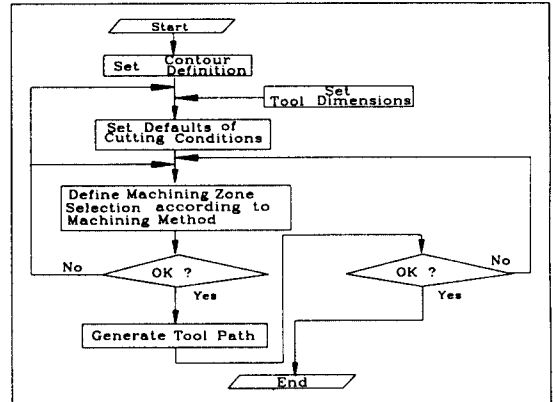


Fig. 18 Algorithm for Tool Path Generation

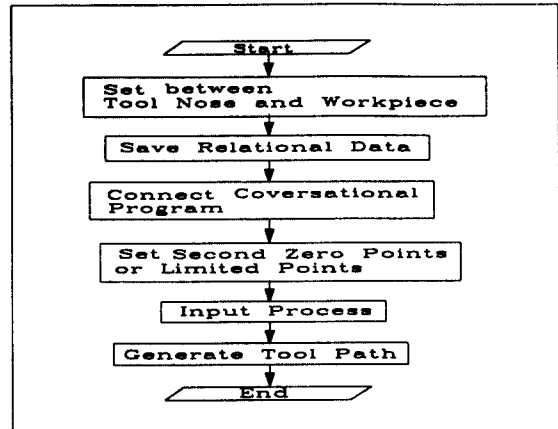


Fig. 19 Practical Flowchart for Tool Path Generation

MMI (Man - Machine Interface) 의 API (Application Programmable Interface) 라이브러리를 통해 실제가공을 위한 공구궤적에 대한 흐름도는 Fig. 19 와 같으며, 공작물의 정확한 장착을 위해 X, Z 축에 평행하게 가공물을 가공한 후, 공구웁셋 데이터를 이용하여 공구와 가공물과의 좌표계 관계를 설정한다.

또한, 공구와 공작물의 충돌회피를 위해 제한구역이 설정되고 Limit Points 를 지정하여 충돌을 방지하는 수단이 설정된다. 자동프로그래밍을 효율적으로 사용하기 위한 방향으로 가공 작업설계의 정확성을 높이는 소프트웨어적인 제한구역 선정작업이 반드시 존재해야 하며, 모의가공시 가공 프로그램이 수행되더라도 기계동작 (X, Z 축 서보모터의 움직임) 이 수행되지 않는 자동프로그래밍 방법을 제안하였다.<sup>(13)</sup> 이에 대한 알고리즘은 다음 Table 5 와 같다.

Table 5 Setting for zone limitation

```

Void Z_axis_Limitation (void)
{
    Double Limit_point_x=0.0, Limit_point_z=0.0 ;
    Double Tool_Offset_x, Tool_Offset_z ;
    structure Final_Shape ;
    Double Machine_Coor, Bite_Coor, Work_Coor ;

    Generation of Window for Limit along to Arrow Key ;
    Display Coordinate System (Machine, Bite, Work) ;
    If (Iteration of Setting Limit for Workpiece is not
    OK) Then
        Set Limitations of Zone ;
    Else
        Endif
    Call Drawing for Final Shape of Workpiece ;
    If (Within Zone Limitation) Then
        Generate Tool Path ;
    Else
        Check Interference between Tool and Workpiece ;
        Stop Tool Path Generation ;
    Endif
    .....
    ..... continued
}
    
```

5.2 가공프로그램 검증

그래픽기반형 프로그래밍 시스템의 경우에 검증 방법은 묘화를 통하여 여러 방향에서 가공물과 공구의 관계를

체크하거나 솔리드형상을 이용하여 최종 가공상태의 공작물을 디스플레이함으로써 검증할 수 있다.<sup>(14,15)</sup> 또한, 와이어프레임 형태로 묘화된 경우에도 궤적의 확대 및 축소를 통해 공구궤적을 조사함으로써 검증할 수가 있다. Fig. 20 은 가공프로그램의 검증 방법으로 화면에 표시한 바와 같이 와이어프레임 (Wireframe) 형태를 보여준다.

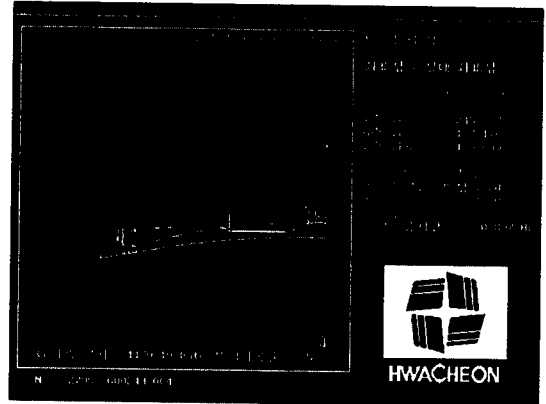


Fig. 20 Graphic Simulation by Wireframe<sup>(16)</sup>

Fig. 21 은 Fig. 6 과 같은 최종형상의 윤곽을 통해 자동프로그래밍 기법으로 G code 가 형성된 과정을 보여주고 있으며, Fig. 20 에 의한 묘화를 통해 검증을 수행한 후 최종적인 가공코드 형성을 보여준다.

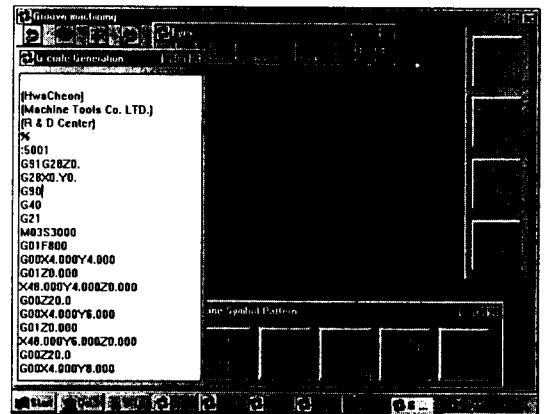


Fig. 21 Screen Display of Generated G code

## 6. 결론

선반용 대화형 프로그램 개발을 통하여 다음과 같은 결론을 얻었다.

1. 개방구조 NC 에 적용할 수 있는 Windows 환경하의 선반용 다기능 대화형 프로그램을 개발하였다.
2. 다목적의 도형데이터 정의, 가공공정을 위한 공구데이터 정의, 그리고 공구와 공작물의 간섭, 충돌회피를 위한 테스트 가공 알고리즘 개발을 통해 가공 프로그램의 단시간 편집 및 작성기법을 제안하였다.
3. Tree 구조 방법론의 도형정의를 통해 모델의 내역 및 공정입력 등의 응용성을 높였다.
4. 열악한 작업환경에서 편리하게 작업할 수 있으며 단순한 부품은 물론 복잡한 부품의 가공을 가능하게 하였다.
5. 아울러 그래픽 기반형 프로그래밍 기법을 사용함으로써 용이한 프로그래밍 방법론을 제시하였고, 작업장프로그래밍 (Shop Floor Programming) 과 CAM 시스템의 혼합을 통해 보다 친숙한 대화형 프로그램 환경을 제안하였다.
6. 개발된 시스템은 사용자의 편의성을 위해 대화형태를 가지며 초보자용 프로그래밍 환경을 중심으로 구축되었다.

## 참고 문헌

1. Yoram Koren, 1983, Computer Control of Manufacturing Systems, McGraw-Hill International editions.
2. Kristin Kelley, 1994, "Getting Beyond G-Code", Modern Machine Shop Magazine, Gardner Publications, Inc.
3. Chris McMahon, Jimmie Browne, 1993, CAD CAM from principles to practice, Addison Wesley Publishing.
4. Paul Kenneth Wright, David Alan Bourne, 1988, Manufacturing Intelligence, Addison Wesley Publishing.
5. Masao Yamamoto, Hirokazu Isobe, Kazuya Yamazaki, 1987, "マシニングセンタ用對話形 MC とまと", 機械と工具, Vol. 12, pp. 123 - 127.
6. FANUC, 1994, FANUC Super Cap Version 1, FANUC LTD., pp. 2-34 - 2-80.
7. Gibbs, 1995, Catalog for Gibbs SFP, Gibbs Co.
8. Byoung K. Choi, 1991, Surface Modeling for CAD/CAM, Elsevier.
9. 김상형저, 1997, 델파이 정복 ver 2.0, 가남사.
10. 日経 CG 社, 1994, CAD の基礎知識, 日経 BP 社.
11. 精密共學會 編, 1991, 生産ソフトウェアシステム, オーム, pp. 89 - 106.
12. 강성균, 이지석, 최종률, 1995, "대화형의 그래픽을 이용한 선삭용 고기능 작업장 프로그래밍 시스템", 한국정밀공학회, 추계학술대회 논문집, pp.707 - 712.
13. 신동수, 정성종, 권영두, 1996, "선반용 대화형프로그램의 개발", 한국공작기계기술학회, 추계학술대회 논문집, pp. 15 - 21.
14. Ki-Yin Chang and Erik D. Goodman, 1991, "A Method for NC Tool path Interference Detection for A Multi-Axis Milling System", Control of Manufacturing Processes, DSC-Vol. 28/ PED-Vol. 52, pp. 23 - 30.
15. 최병규, 1989, Cam 시스템과 CNC 절삭가공, 청문각.
16. 신동수, 1995, CAMMAN User's Manual, 화천기공, pp. 21 - 24.