

# 금형가공을 위한 공정설계시스템

Process Planning System for Machining of Mold Dies

## 금형부품 가공용 작업설계 시스템 개발

Development of the Operation Planning System  
for Mold Die Manufacturing

연구기관  
한국과학기술연구원

과 학 기 술 처

# 제 출 문

## 과학기술처 장관 귀하

본 보고서를 "금형가공을 위한 공정설계시스템" 과제의 (세부과제 "금형부품 가  
공용 작업설계 시스템 개발"의) 제 2차년도 최종보고서로 제출합니다.

1992. 10.

주관연구기관명 : 한국 과학 기술 연구원(KIST)

총괄연구책임자 : 노 형 민 (CAD/CAM 연구실)

연구 책임자 : 박 면 웅 (CAD/CAM 연구실)

연구 원 : 이 준 성 (CAD/CAM 연구실)

협동연구기관명 : 서울 산업 대학교 기계설계과

협동연구책임자 : 맹 회 영(산업대학교 기계설계과 교수)

연구 원 : 김 성 근( 기계설계과 대학원)

서 창 길( 기계설계과 대학원)

호 춘 기( 기계설계과 대학원)

여 백

# 요 약 문

## I. 제목

금형부품 가공용 작업설계 시스템 개발

## II. 연구개발의 목적 및 중요성

상자형 가공물 특히 금형부품상의 형상정보가 형상특징 (feature) 단위로 주어졌을 때 이를 실제로 절삭가공 하기 위하여 필요한 절삭 파라미터 들을 자동으로 산출할 수 있는 소프트웨어를 개발하여 설계정보를 가공정보로 자동으로 변환함으로써 CIM 체제내의 정보의 변환 및 흐름을 원활 하게 하는 것을 목적으로 하였다. 최종년도에는 절삭양식별로 절삭성에 관련된 논리들을 지혜형으로 공리화하고, 이를 통해 종합화된 구조의 절삭가공 데이터베이스를 구축함과 동시에 이를 효과적으로 활용할 수 있게 하였다.

공정설계를 중심으로 그 전후 기능과의 연계가, 통상 CIM을 추구하는데 가장 큰 장애요소인데, 공정설계의 front interface인, 설계 data로부터 자동 가공형상인식 system이라든지, 가공방법의 합리적 공리화 과정, 효율적인 절삭성 데이터의 확보와 참조 등은 CAD와 CAM을 통합 (integration) 하는데에 필수적인 기능이기도하다.

공작기계 및 공구종류가 결정된후 개별형상을 절삭가공하는데 필요한 parameter들(공구치수,가공조건등)을 산정하는 기능을 작업설계라고 할 수 있는데, 가공parameter를 결정하기 위한 knowhow는 일부 정리되어 있기도 하고 engineer의 체험으로 저장되어 있기도 하여 작업설계에 반영하고 있으나 이들을 취합, software화 하는 것이 필요하다. 산재된 현장 knowhow의 체계적 수집, 효율적인 data structure 설계 등에 어려움이 있어 그런 data base를 구축한 기업체가 거의 없다.

### III. 연구개발의 내용 및 범위

금형설계 시스템에서 출력된 형상정보와, 공정설계 시스템에서 설정된 공정정보를 입력으로 하여, 각 공정에서 필요한 절삭파라미터들을 산출하는 작업설계 시스템 (COPS ; Computer aided Operation Panning System) 개발하였다.

공정설계에서 설정된 기능 feature 들을 8개의 절삭가공 feature 로 구분 (face 밀링, side밀링, pocket밀링, slot밀링, 구멍가공, 나사치기, 카운터 보링, 카운터싱킹)하여 feature에 대한 정보를 데이터화일을 통하여 받아들이고, feature 형상정보와 공작기계용량, 가공물 재질, 공구종류가 주어졌을때, 그 feature 를 가공하기 위한 공구 사이즈를 선택하고, 그공구를 사용할때 절삭속도, 이송속도, 절삭폭, 절삭깊이를 결정하여 출력하도록

하였다. 이 목적을 위하여 재질화일, 공작기계 화일, 공구 화일, 절삭조건 화일, 절삭가공 데이터베이스화일을 구성하고 이를 유지관리하기 위한 소프트웨어를 별도로 개발하였다.

작업설계의 핵심이되는 절삭가공 데이터베이스를 구축하기 위해서 최종년도에는 다음의 내용들이 수행되었다.

- (1) 절삭성 데이터의 구축을 위한 모델링 실시
- (2) 국내에서 쓰고 있는 금형용 재질과 금형가공을 위한 절삭양식에 사용되는 공구의 정리 후 분류 코딩
- (3) 현장작업자 중심의 실용 절삭성 데이터 조사 확보
- (4) 절삭양식별 알고리즘 정리와 알고리즘을 실시간내 근사추론 시킬 수 있는 기법 탐색
- (5) 전문가를 통해서 각 feature 별로 수집된 절삭성 데이터들을 입력으로 각각의 절삭 파라미터들에 대한 클래스마다의 절삭계수들을 자동결정케 하는 editor를 개발, 이를 통한 절삭계수 화일 구축, 절삭계수 화일 데이터의 신뢰성 증가를 위해 fuzzy 처리
- (6) 데이터베이스 install할 때에는 메인모듈에서 사용하고 있는 각각의 절삭 파라미터들이 절삭조건 결정 프로그램의 pre-processor 모듈로 연결될 수 있도록 인터페이스 프로그램 작성

#### IV. 연구개발결과 및 활용에 대한 건의

절삭실험을 통하여 산출된 절삭조건이 적정함이 확인되었고 설계시스템 및 공정설계와의 정보화일을 통한 정보흐름이 원활히 수행되어 전체 CIM환경에서의 총합 생산성을 높일 수 있을 것으로 사료된다.

각 사업장의 환경에 따라, 화일 관리 소프트웨어를 사용하여 지원 데이터화일을 작성하여, 사업장에 적합한 작업설계를 수행할 수 있고, 화일 내용을 지속적으로 개선, 보완 함으로써 가공노우하우의 축적을 이룰 수 있다.

본연구에서 얻어지는 공리화 알고리즘이나 신경회로망의 Hidden layer, Fuzzy 처리과정, 절삭계수 DB editor 등은, 막연한 경험데이터들을 실용식으로 표현하는 방편으로도 활용할 수 있을 것으로 사료된다.

# SUMMARY

## I. Title of the Project

Development of the Operation Planning System for Mold Die Manufacturing

## II. Objective and Significance of the project

The developed system called COPS (Computer aided Operation Planning System) generates cutting conditions based upon database and reasonable logic facilitated in a software system to complement process planning system which determines machining operation, selects machine tool and cutting tool, and decides the operation sequence. The system plays the roll of bridge between design and machining, which enables the uninterrupted flow and automatic conversion of information, as well as the roll of a tool for less experienced engineer to carry out consistant and optimum operation planning.

Process planning function is still the bottleneck for information flowing uninterruptedly from design to machining process in CIM environment. Process planning can be devided to process design and operation planning. The former function selects machinning process, machine tools, decide the sequence of operations, and performs capacity planning. The latter, which usually relys on machine operator's experience, decides



machining parameters - tool size, depth and width of cut, cutting speed, and feed rate.

Decision of cutting parameters has been neglected as a research theme, maybe because a logical system needs too many on-site factors to be considered to produce practical output. More effort has been given to other area of software development, and the decision of machining parameters remains as a manual operation. Algorithms representing metal cutting procedure and effective machinability data are necessary in order that workman's knowhow may be accumulated in a recorded form to keep improved and to be passed down to less-experienced machinist.

### III. Contents of the project

An operation planning system named COPS which generates cutting parameters - tool size, depth and width of cut, cutting speed, and feed rate - with input of geometric information from parametric design system and process information resulted from process planning system has been developed.

The functional features assigned to machining operations are categorized to 8 machining features - face milling, pocket milling, slot milling, hole making, tapping, counter boring, and counter sinking. Algorithms to carry out operation planning for those categories according to the shape of each feature, capacity of selected machine tool, workpiece material, and cutting tool, were established and implemented. For supporting this software, five data files have been designed, - material file, machine tool file, cutting

tool file, sample cutting condition file, and cutting technology file - and a software program files has been made for the maintenace of those data files.

Following items are carried out in order to establish metal cutting data base the kernel of the improved operation planning which is developed in the final year.

- (1) Modelling of metal cutting process parameter, process state and process control parameter
- (2) Classification and coding of materials for mold die manufacturing and cutting tools used for machining operations of mold die components
- (3) Acquiring practical machinability data from field experts
- (4) Preparation of algorithms of metal cutting logic for each category of cutting operations, and investigation of AI technique to approximate the algorithms in real time
- (5) Development of cutting parameter editor by which cutting condition data file is created through automatic decision of cutting parameters from machinability data gathered from the experts, and fuzzy processing of cutting parameter data for increased reliability
- (6) Coding interface program which relates cutting parameters to pre-processor module of cutting codition selection program when installation of the prepared database

#### IV. Results and proposals for the future

The output of the established system has been proved practical through metal

cutting test. Total efficiency of CIM system can be enhanced by the uninterrupted flow of information from design to machining.

The field knowhow can be accumulated and improved by maintain the data files continuously. Each machine shop can reflect its policy and environment to the system by establishing own data files using the maintenance program.

The functions obtained through this research project - modelling of metal cutting process, hidden layer of neural network, fuzzification proceduc, and cutting parameter DB editor could be used as a method to express abstract emperical data in practical logics.

# Contents

Chapter I	Introduction	19
Section 1	Backgrounds of the research	19
Section 2	State of the art	21
1.	Overseas	21
2.	Domestic	22
3.	Industry	22
Section 3	Research contents and scope	24
Section 4	Composition of this report	28
Chapter II	Description on the developed system	29
Section 1	Structure of COPS system	29
Section 2	Design - process planning - operation planning interface	31
Section 3	Structure and function of the operation planning module	34
1.	Milling process	40
2.	Hole making process	45
3.	Threading process	45
4.	Conterboring/countersinking	45
Section 4	Supporting data file	46
1.	Material file	46
2.	Machine tool file	46
3.	Cutting tool file	46

	4.	Cutting condition file -----	46
	5.	Cutting Technology File -----	47
	6.	File maintenance file -----	47
	Section 5	Input/output format -----	49
Chapter III		Formulation of relationship among process parameters of cutting state and establishment of mapping -----	53
	Section 1	Decision of process parameter -----	53
	1.	Introduction -----	53
	2.	Factors affecting cutting state -----	55
	3.	Mapping -----	58
	4.	Data structure in mapping -----	60
	Section 2	Investigation on methods to establish mapping relation	64
	1.	Introduction -----	64
	2.	Multi layer perceptron model -----	65
	3.	CMAC neural network -----	68
	4.	Rules to control stochastic data by fuzzy number -----	70
Chapter IV		Development of DB Editor and Database construction -----	73
	Section 1	Acquisition of machinability data -----	73
	1.	Establishment of feature class -----	73
	2.	Questionare preparation and data acquisition -----	86
	Section 2	Programming of DB Editor -----	94

	1.	Construction of Multi-layer perceptron -----	94
	2.	Structure of data input of neural network3 -----	102
	3.	Learning process of neural network -----	106
Section 3		Construction of cutting parameter database -----	107
	1.	Database file system -----	107
	2.	Data access through fuzzy number process -----	107
	3.	Modification of data by rules -----	110
Chapter V		Conclusion -----	118
Section 1		Summary -----	118
Section 2		Discussion on the result of the research -----	119
References		-----	121
Appendix I.		Sample cutting parameter files for individual feature -----	125
Appendix II.		Example of executing DB Editor and List of result -----	131

여 백

# 목 차

제 1 장	서 론	19
제 1 절	연구 배경	19
제 2 절	연구 사례의 조사	21
1.	외국의 경우	21
2.	국내의 경우	22
3.	산업계 현황	22
제 3 절	연구 내용 및 범위	24
제 4 절	보고서의 구성	28
제 2 장	개발된 시스템의 개요	29
제 1 절	COPS 시스템의 구성	29
제 2 절	설계-공정설계-작업설계 인터페이스	31
제 3 절	작업설계 메인 모듈의 구성과 기능	34
1.	밀링공정	34
2.	구멍가공	40
3.	나사가공	45
4.	Counterboring / Countersinking	45
제 4 절	지원 데이터베이스 file	46
1.	금형용 재질 file	46
2.	공작기계 file	46
3.	절삭공구 file	46
4.	절삭조건 file	46



5.	절삭기술 file	47
6.	Data file 보수 유지 프로그램	47
제 5 절	입출력 양식	49
제 3 장	절삭가공인자의 공리화 및 매핑관계의 구성	53
제 1 절	절삭가공인자의 결정	53
1.	개 요	53
2.	가공상태에 영향을 미치는 인자	55
3.	매핑관계의 구성	58
4.	매핑관계의 기본데이터 구조	60
제 2 절	매핑관계의 구성방법론	64
1.	개 요	64
2.	다층 퍼셉트론 모델	65
3.	CMAC 신경회로망	68
4.	퍼지숫자에 의한 불확실성 처리 규칙	70
제 4 장	DB Editor의 작성과 데이터베이스의 구축	73
제 1 절	절삭성데이터의 수집방법	73
1.	Feature별 class의 구성방법	73
2.	설문작성 및 수집데이터	86
제 2 절	DB Editor의 프로그래밍	94
1.	Multi-layer 퍼셉트론의 구성	94
2.	신경회로망 데이터입력 파일구조	102
3.	신경회로망의 학습과정	106
제 3 절	절삭가공데이터베이스의 구축	107
1.	데이터베이스 화일시스템	107
2.	Fuzzy값 처리에 의한 데이터 액세스	107

3. 규칙에 의한 데이터의 갱신	110
제 5 장 결 론	118
제 1 절 요약	118
제 2 절 연구 결과의 토의	119
References	121
부록 I Feature별 표준절삭조건 화일 리스트	125
부록 II DB Editor의 수행 예시와 출력결과 List	131

여 백

# 제 1 장 서 론

## 제 1 절 연구 배경

일반적으로 금형부품의 생산시스템은 설계, 공정계획, 가공, 조립, 품질 관리, 정보운용 및 구매 등의 여러가지 활동이 서로 연결된 복합구조로서 전형적인 다품종 소량생산 형태를 갖는다. 이같은 특성 때문에 금형산업은 유연성 자동화 시스템의 적용이 매우 효과적으로 응용될 수 있는 대상이며, 국내에 있어서도 1980년대에 들어서면서 CAD/CAM 시스템이 금형제조 산업에 본격적으로 소개되어 사용되었고, 현재에는 제품설계, 설계해석, 공정설계, NC 프로그래밍, 생산계획 및 관리, 측정 및 검사 등 거의 모든 분야에 걸친 소프트웨어들이 보급되어 기술부문에서의 불가결의 설계 툴로 자리잡아 가고 있다.

그러나 국내의 금형업체들에 있어서는 하드웨어와 소프트웨어는 어느 정도 필요성을 인식하여 도입하고는 있으나, 이들 시스템을 효과적으로 사용할 수 있는 휴먼웨어와 어플리케이션웨어의 수가 절대적으로 부족하고 이들 시스템의 기초가 되는 현장 노하우와 경험 데이터의 구축이 제대로 되어 있지 못하다. 또한 뛰어난 기능을 갖춘 단위 CAD/CAM 시스템이 있다고 하더라도 각 기능들 사이의 업무나 정보들이 재래식 방법으로 독립 운용되기가 일쑤여서 컴퓨터의 역할은 업무보조의 성격으로 밖에 이용되지 못하고 있고 자동화 성향의 면에서도 매우 불완전한 양상을 갖게 하고 있다.

따라서 이 부문에서 직면하고 있는 기술인력의 확보, 원가 절감, 생산성 제고, 품질향상, 기술 축적 등의 제반 문제들의 해결을 위해서는 현장 노하우의 체계적인 축적 및 적용과 함께 각각의 단위 기능요소들을 통합하여

자동화 성향을 높이고 손쉽게 시스템을 운용할 수 있게 하는 것이 큰 과제라 하겠으며, 이를 위해서는 적극적으로 지혜형 컴퓨터를 활용하여 종합적인 CIM 시스템을 구축하고 운용될 때 그 효율이 극대화 될 것이다.

그리고 이들 부문의 CIM을 구축하는데 가장 큰 장애요소로는 공정설계를 중심으로 한 그 전후 기능들과의 연계 문제인데, 공정설계의 front interface 대상인 가공형상의 인식시스템을 통하여 절삭가공에 필요한 파라미터들을 도출하는 작업설계 시스템은 CAD와 CAM을 통합(Integration) 하는데 필수적인 기능을 담당하고 있다. 공작기계 및 공구 종류가 결정된 후 개별 형상을 절삭가공하는 데에 필요한 parameter들 (공구치수, 가공조건 등)을 산정하는 기능을 작업설계라고 할 수 있는데, 가공 parameter를 결정하기 위한 knowhow는 일부 정리되어 있기도 하고 engineer의 체험으로 저장되어 있기도 하여 작업설계에 반영하고 있으나 이들을 취합, software화 하는 것이 필요하다. 산재된 현장 knowhow의 체계적 수집, 효율적인 data structure 설계 등에 어려움이 있어 그런 data base를 구축한 기업체가 거의 없다. 작업설계시스템의 핵심인 가공방법의 합리적 공리화 과정, 효율적인 절삭성 데이터의 확보와 참조 등은 프로세스를 지배하는 인자들이 매우 다양하면서도 논리화 되지 않는 것이 많기 때문에 형상론적인 접근방법으로 적용하는 경우 이외에는 주로 경험 또는 감각에 의존하여 설정되는 경우가 대부분이다. 그 때문에 지금까지 많은 연구가 실시되어 왔지만 아직도 신뢰도와 예측 정확도의 면에 있어서는 현장에의 적용 구비조건에 미흡한 실정이다. 그러므로 이를 해결하기 위해서는 앞으로 AI 기법을 응용한 광범위하고 보다 보편화된 CAPP용 데이터베이스의 확보가 절실히 요구되며 구축된 데이터베이스를 손쉽게 탐색하여 작업조건을 적절히 결정할 수 있는 도구들이 필요하다고 보겠다.

## 제 2 절 연구사례의 조사

### 1. 외국의 경우

설계 및 가공의 통합은 공정설계를 중심으로 그 전후 기능과의 interface를 통하여 추구하고 있다. 공정설계의 기능은 그림 1-1과 같이 협의의 공정설계와 작업설계로 나눌 수 있다. 작업설계에서는 실제 절삭가공에 필요한 parameter들 (공구치수, 가공조건 등)을 산정하고 tool path를 생성하므로 공정설계의 down stream interface라고 할 수 있다. 공정설계의 front interface인, 설계 data로부터 자동 가공형상인식기법은 독립적으로 많은 연구가 진행중이나 작업설계는 대개의 경우 공정설계시스템 기능의 일부로서만 언급되어 왔다.

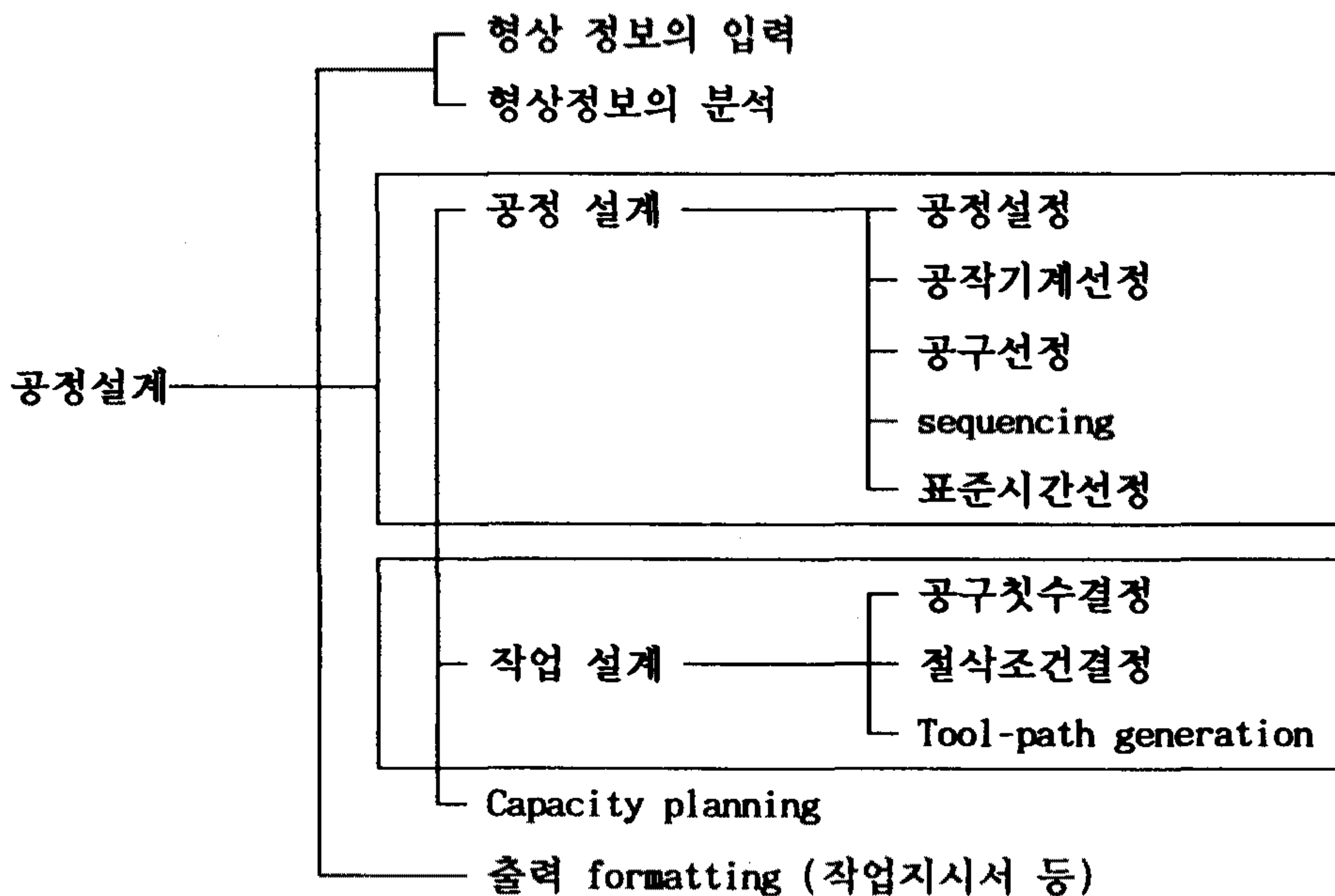


그림 1-1. 공정설계 기능구분

그림 1-2는 외국에서 개발되어온 공정설계시스템 중 절삭 parameter들도 생성하는 시스템들을 보여주고 있다. 많은 연구가 진행되어 온 것에 비하여, 실질적으로 산업현장에서 활용되고 있는 시스템은 소수이다.

## 2. 국내의 경우

KIST CAD/CAM연구실에서는 공정관리 프로그램, 자동생산정보 시스템, 회전형상 부품에 대한 절삭가공 가공공정 및 가공기계 선정 프로그램 등을 개발한 실적이 있다. 서울대에서는 expert기법을 사용한 (협회의)공정설계 시스템을 개발하였고, 부산대학에서는 사출금형가공용 공정설계시스템을 개발하였다. KIMM에서는 현재 press 금형가공용 공정설계시스템을 개발중에 있다. 그러나 아직 실용화된 시스템은 없고 작업설계기능을 가진 공정설계시스템은 전무한 형편이다.

## 3. 산업계 현황

설계 system, CNC 공작기계, analysis software, NC programming system 등은 구입, 사용할 수 있는 상용 system이 존재하여 CAD/CAM 도입 역사가 일천한 한국산업체들도 필요에 따라 완전하지는 않지만 단위 system으로 구입하여 활용을 하고 있고 CIM을 지향하는 노력의 시작점으로 삼고 있다.

그러나 설계와 가공의 bridge 역할을 담당하는 공정설계 분야는 product 및 workshop dependent한 요소가 많아 구입가능한 상용 software가 거의 없다고 하더라도 유용하게 사용하기 위해서는 user가 구입후 system자체를 전면 개편하여야 한다. CAPP의 front interface인, 설계 data로부터 자동 가공형상인식system은 integration을 위해 필수적 기능이어서 많은 연구가 진행중이나 paper등을 통해 발표된 기법을 기업체가 활용하기에는 시간, 연구인력이 부족하다. 가공 parameter를 결정하기 위한 knowhow는 일부 정



리되어 있기도 하고 engineer의 머리속에도 저장되어 있기도 하여 공정 및 작업설계에 반영하고 있으나 이들을 취합, software화 하는 것이 필요하다. 산재된 knowhow의 체계적 수집, 효율적인 data structure설계 등에 어려움이 있어 그런 database를 구축한 기업체가 거의 없다.

System Name	Part Shape	Planning Approach	Automated Function			
			sequence	m/c & tool selection	cutting parameter	tool path
AUTAP	Rotational & Sheet metal	Decision table	0	0	0	0
AUTOPLAN	Rotational	Variant	0	0	0	
APPAS	Prismatic	Decision tree	0	tool only	0	
CAPP	Rotational	Decision table	0	0	0	
EXAPT	Rot. & Prism	N / A			0	0
XPS-1	Rot. & prism	Variant * D/T	0	0	0	
CAPSY	Rotational	Decision table	0	0	0	0
ICAPP	prismatic	Decision tree	0	0	0	0

그림 1-2. 작업설계기능을 가진 공정설계시스템



### 제 3 절 연구 내용 및 범위

금형설계 시스템에서 출력된 형상정보와, 공정설계 시스템에서 설정된 공정정보를 입력으로 하여, 각 공정에서 필요한 절삭파라미터들을 산출하는 작업설계시스템(COPS: Computer aided Operation Panning System)이 개발되었다. 그 전후 기능과의 관계는 그림 1-3과 같다.

공정설계에서 설정된 기능 feature들을 8개의 절삭가공 feature로 구분 (face밀링, side밀링, pocket밀링, slot밀링, 구멍가공, 나사치기, 카운터보링, 카운터싱킹) 하여 feature에 대한 정보를 데이터화일을 통하여 받아들이고, feature형상정보와 공작기계용량, 가공물재질, 공구종류가 주어졌을 때, 그 feature를 가공하기 위한 공구 사이즈를 선택하고, 그 공구를 사용할 때 절삭속도, 이송속도, 절삭폭, 절삭깊이를 결정하여 출력하도록 하였다.

이 목적을 위하여 재질화일, 공작기계 화일, 공구화일, 절삭조건 화일, 절삭가공 데이터베이스 화일을 구성하고 이를 유지 관리하기위한 소프트웨어를 별도로 개발하였다.

작업설계 과제의 2차년 ( 91.10 ~ 92.9 )에는 1차년에 기본적인 골격을 갖춘 시스템을 보완 확장하였다. 특히 단순한 핸드북 타입의 절삭조건 화일을 개선하여 절삭 양식별로 절삭성에 관련된 논리들을 지혜형으로 공리화하고, 이를 통해 종합화된 구조의 절삭가공 데이터베이스를 구축함과 아울러 이를 효과적으로 활용할 수 있게 하였다.

이를 위해 절삭성 데이터베이스의 구축 연구는 다음의 6단계 내용을 거쳐 수행되었다.

#### (1) 절삭성 데이터의 모델링과 COPS의 형상정보 파라메타의 설정

각주형 금형부품의 절삭가공에 관련된 공정계획 방법론, 기존의 개발된

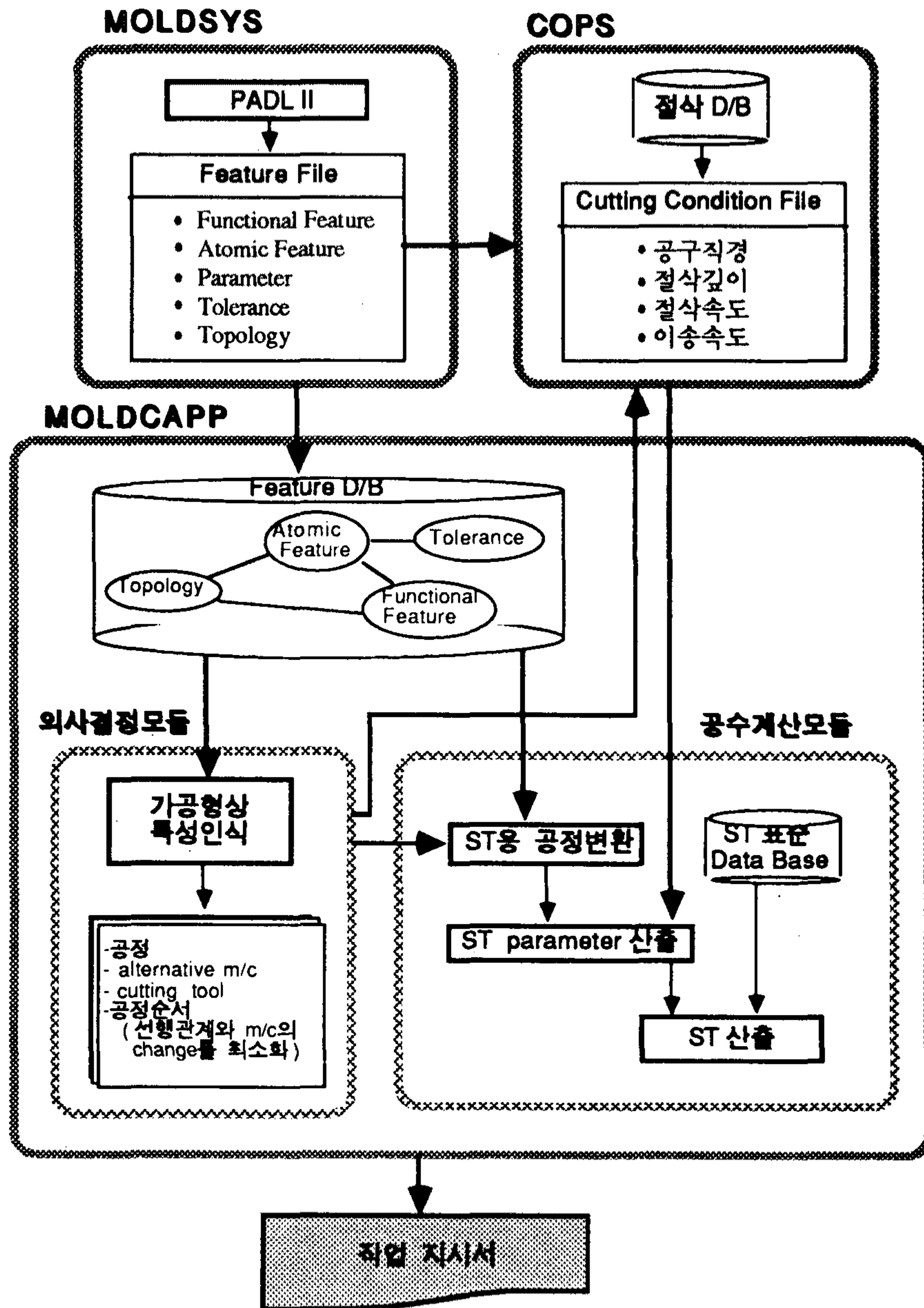


그림 1-3. MOLDSYS - MOLDCAPP - COPS - ST 의 관계

CAPP 에서의 데이터베이스 구축에, 절삭성에 대한 핸드북, 국내 금형업체들의 경험 데이터 등을 중심으로 절삭성 데이터들을 확보하였고 이를 통해 절삭성데이터의 구축을 위한 모델링을 실시하였으며, 이들 과정에서 절삭성 데이터들과 형상정보들과의 상호관계를 기초적인 2단계의 매핑관계로 구성하였다.

#### (2) 금형용 재질과 사용커터들의 조사 및 분류

국내에서 사용하고 있는 금형용 공작물 재질들을 조사하여 이를 절삭성과 구입선, 열처리정도 등을 기준으로 분류 및 코딩하였고, 또한 각주형 공작물의 가공에 필요한 절삭양식들에 대하여 사용 커터들을 조사한 후, 이를 공구재료와 공구의 기하학적인 특징, 제조 메이커, 정적 동적 특성, 치수 및 연삭성 등을 토대로 정리한 후 분류 코딩하였다.

#### (3) 절삭성 데이터의 확보 및 정리

가공물재질과 사용커터의 분류 및 코딩이 완료된 후에는 모델링된 클래스에 따라서 절삭성 데이터의 확보를 위한 설문양식을 작성하였으며, 이를 통해 핸드북의 범위를 넘어서 현장작업자 중심의 실용 절삭성 데이터들을 주어진 샘플 모델에 대해 앙케이트 조사를 실시하였다. 이들 조사는 2차의 단계를 거쳐서 수행이 되었는데, 1차 조사에 있어서는 주로 가공시에 중요시되는 인자의 파악을 위한 조사였고, 2차 조사에서는 현장에서 경험이 풍부한 전문가를 대상으로 한 표준절삭조건의 수집을 위한 조사였다.

#### (4) 절삭성 이론의 공리화 및 가능 AI 기법의 탐색

각 절삭양식에 있어서는 응용가능한 절삭성 이론들을 바탕으로 수치해석 또는 공리화가 가능한 알고리즘들만을 추출하여 정리하였고 이들을 실용화된 선형식으로 변형시켰다. 또한 기존의 Expert system 에서 응용되고 있는 방법론들을 중심으로 공리화된 절삭성 알고리즘을 실시간 내에서 근사 추론시킬 수 있는 기법들을 탐색하였으며, 이를 수치적인 모델러를 통해서 적용

가능성을 검토하였다.

이에 따라 본 연구에서는 Regression 방법, 다중 퍼셉트론 방법, CMAC (Cerebellar Model Articulation Controller) 방법, GMDH (Group Method of Data Handling) 방법 등의 4가지 방법에 대해서 검토하게 되었으며, 이들 중 절삭성 데이터의 특징인 가식적 데이터의 처리문제, 실용적 데이터의 구축 과정에서의 수렴성, 가공 파라메타들의 비중도 정의 및 단위화(normalize) 등의 문제들에 있어서 다중 퍼셉트론 방법과 CMAC 방법이 초과적으로 사용될 수 있음을 검토할 수 있었다.

#### (5) 표준데이터 화일 DB editor의 작성 및 데이터베이스의 구축

데이터베이스의 각 요소들을 자동으로 배치 구성될 수 있도록 하는 DB editor를 절삭성 알고리즘과 위의 2가지 AI 룰들을 기초로 프로그래밍 하였으며, 5인의 전문가를 통해서 각 feature 별로 수집된 절삭성 데이터들을 입력으로 각각의 절삭 파라메타들에 대한 클래스마다의 절삭계수들을 자동 결정케 하는 DB editor를 개발하였고, DB editor의 합리성이 확인된 후에는 본격적으로 이를 실행시켜 main module에서 사용될 표준 절삭계수 화일들을 구축하였다.

#### (6) Main module과의 연결과 테스트 런

5인의 전문가 데이터등에 대해 작성된 각각의 절삭계수 화일들은 Fuzzy처리를 통하여 데이터들의 신뢰성을 보강시켰으며, 완성된 표준 절삭계수 데이터베이스들은 COPS의 내장된 module 위치에 install하였다. 그리고 이들 데이터 베이스들을 install할 때에는 COPS에서 사용되고 있는 각각의 절삭 파라메타들이 절삭조건결정 프로그램의 pre-processor 모듈로 연결될 수 있도록 인터페이스 프로그램을 작성하였으며, 또한 절삭조건 결정 프로그램의 출력값들이 main module에서의 common데이터들로 사용될 수 있도록 하는 부가 프로그램도 작성하였다.

## 제 4 절 보고서의 구성

금형가공의 작업설계 연구는 90년 10월에 CIM 사업에 추가되어 그 첫해에 작업설계 알고리즘의 개발, 지원데이터베이스의 구조설계, 관련 데이터베이스의 구축이 수행되었는데, 시스템의 주요기능을 갖추게 한 1차년 개발 내용은 제 2장에 간략하게 기술되었다.

91년 10월부터 수행된 2차년도 연구는 알고리즘상 문제가 있었던 부분의 수정과 절삭성 데이터베이스의 개선에 중점을 두었는데, 절삭가공데이터의 모델링, 절삭성이론의 공리화, 표준데이터화일 에디터 작성 등이 연구되었고 이 결과와 메인 모듈과의 인터페이스가 수행되었다.

제3장과 4장에는 절삭성 데이터베이스의 개선에 관련되는 내용들이 수록되었다.



## 제 2 장 개발된 시스템의 개요

### 제 1 절 COPS 시스템의 구성

COPS(Computer aided Operation Planning System)시스템의 메인프로그램과 공정별 서브루틴은 FORTRAN77으로 프로그램되었고, 여기에 C-언어로 작성된 인터페이스 모듈을 통하여, 형상정보를 포함한 공정정보를 데이터 파일로 입력시키면 각 절삭공정에서 필요로 하는 가공데이터들이 출력되도록 하였다. 산출된 절삭조건은 필요한 경우 시스템 사용자에게 의해 확인 및 수정할 기회가 주어진다.

COPS시스템의 구조와 입출력은 그림 2-1과 같다. 제 2절에는 CAD, 공정 설계로부터 데이터가 입력되는 과정이 기술되고 있고 제 3절에서는 작업설계 메인모듈이 공정별로 수행되는 과정이 설명되며 제 4절에서는 지원데이터파일의 구성과 관리에 관한 내용이 기술된다. 그리고 절삭조건 데이터베이스를 개선, 확장한 연구내용은 제 3장과 4장에 별도로 기술된다.

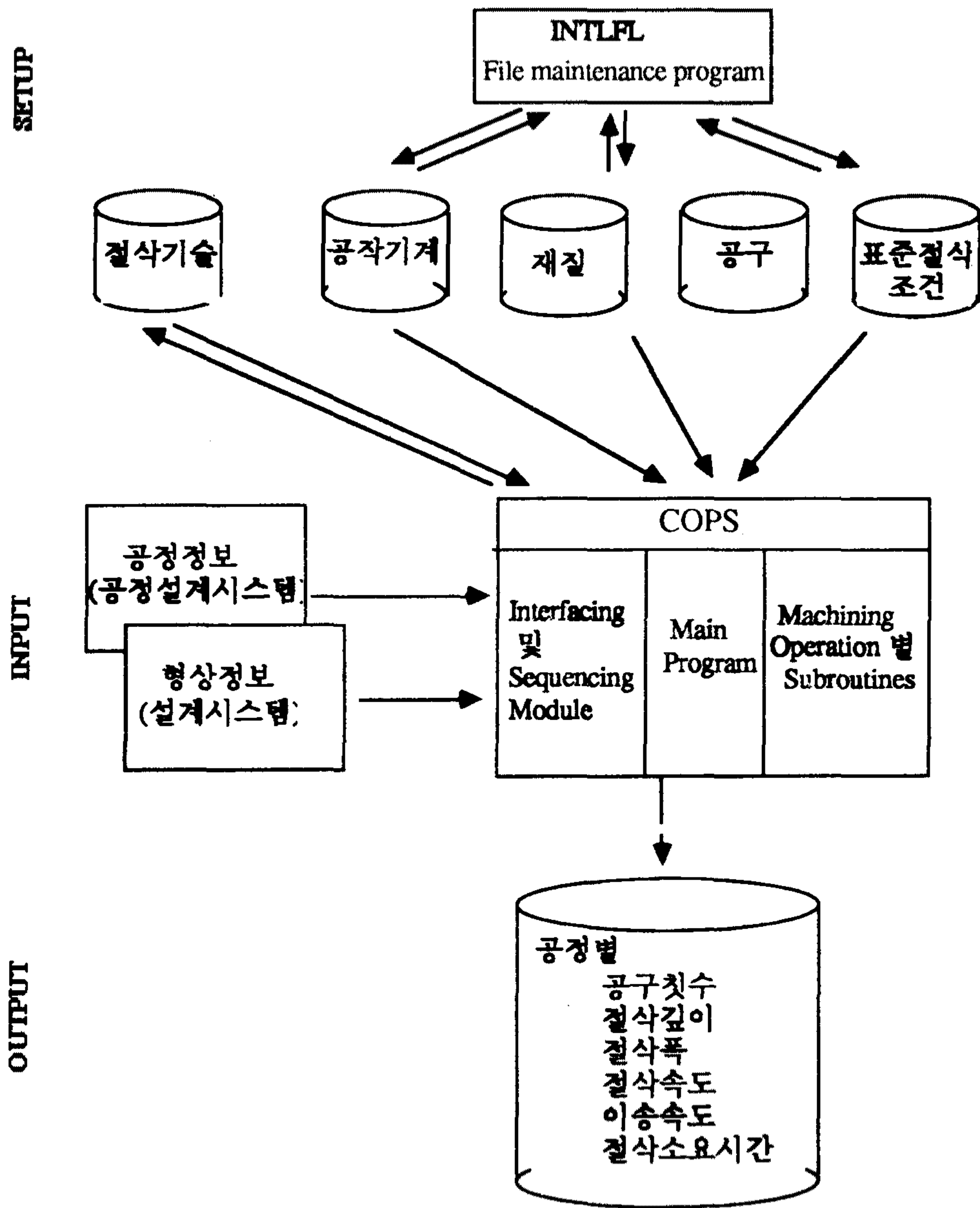


그림 2-1. 작업설계 시스템의 구성

## 제 2 절 설계-공정설계-작업설계 인터페이스

사출금형 전용설계 시스템(Moldsys)은 기본형상으로 26개의 atomic feature를 사용하여 금형부품들을 설계한다. 각 부품은 atomic feature들의 조합인 기능feature들을 기본으로 형상정의되어 화일의 형태로 공정설계시스템 (Moldcapp)과 작업설계시스템(COPS)로 전송한다. 데이터파일은

1. Set.data -> 몰드세트에 관한 정보
2. Part.dat -> 부품에 관한 정보
3. Fnfr.dat -> Functional feature에 관한 정보
4. Atfr.dat -> Atomic feature에 관한 정보
5. Geom.dat -> 기하학적데이터에 관한 정보

로 구성되어 있다.

공정설계의 출력 공정은 functional feature와 多對一 (many to one)의 관계를 가지고 있다. 상기 정보가 linked list 구조로 변환되어 저장되기 때문에, 각공정에 해당하는 atomic feature를 추적하여 형상정보를 추출할 수 있다. 이는 C-언어로 프로그램된 인터페이스 모듈에 의하여 수행된다.

작업설계에 필요한 형상데이터를 비롯한 제 정보를 일정한 장소에 저장토록 하여 작업설계시 입력데이터로 활용하며 또 한편으로 각각의 atomic feature에 대한 세부 sequence를 정한다.

이 C-언어 모듈은 COPS의 메인 프로그램으로서 입력된 형상 정보 데이터 - 26개의 atomic feature로 정의된 - 를 작업설계 시스템 COPS의 8개 가공 feature중의 하나로 매핑하여 해당되는 작업설계 subroutine를 call하고 필요한 데이터를 공급한다. 설계 단위형상 - 가공 형상 특징- 절삭가공 공구간의 대응 관계는 그림 2-2와 같다.



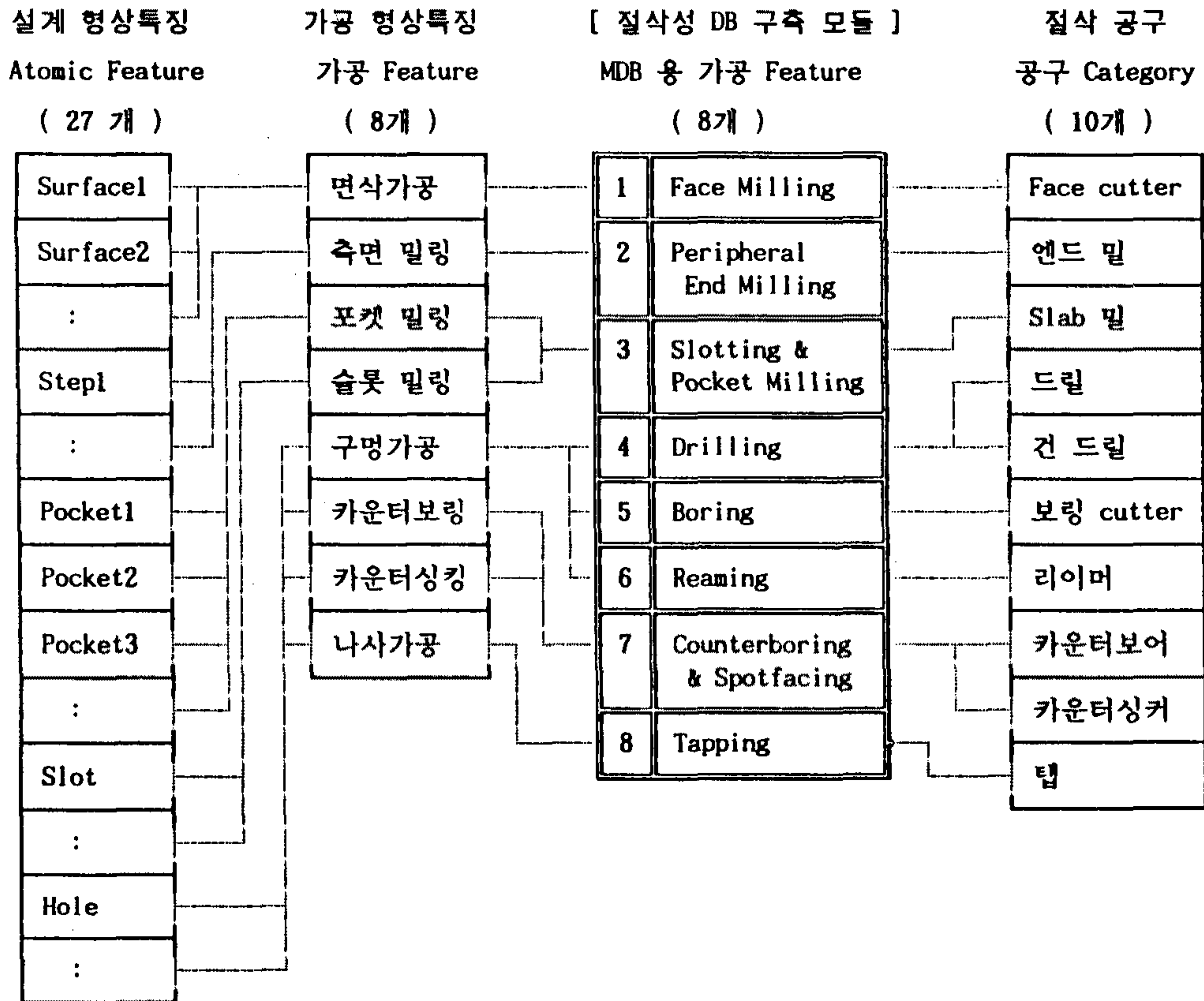


그림 2-2. 형상정보의 절삭가공으로의 연계

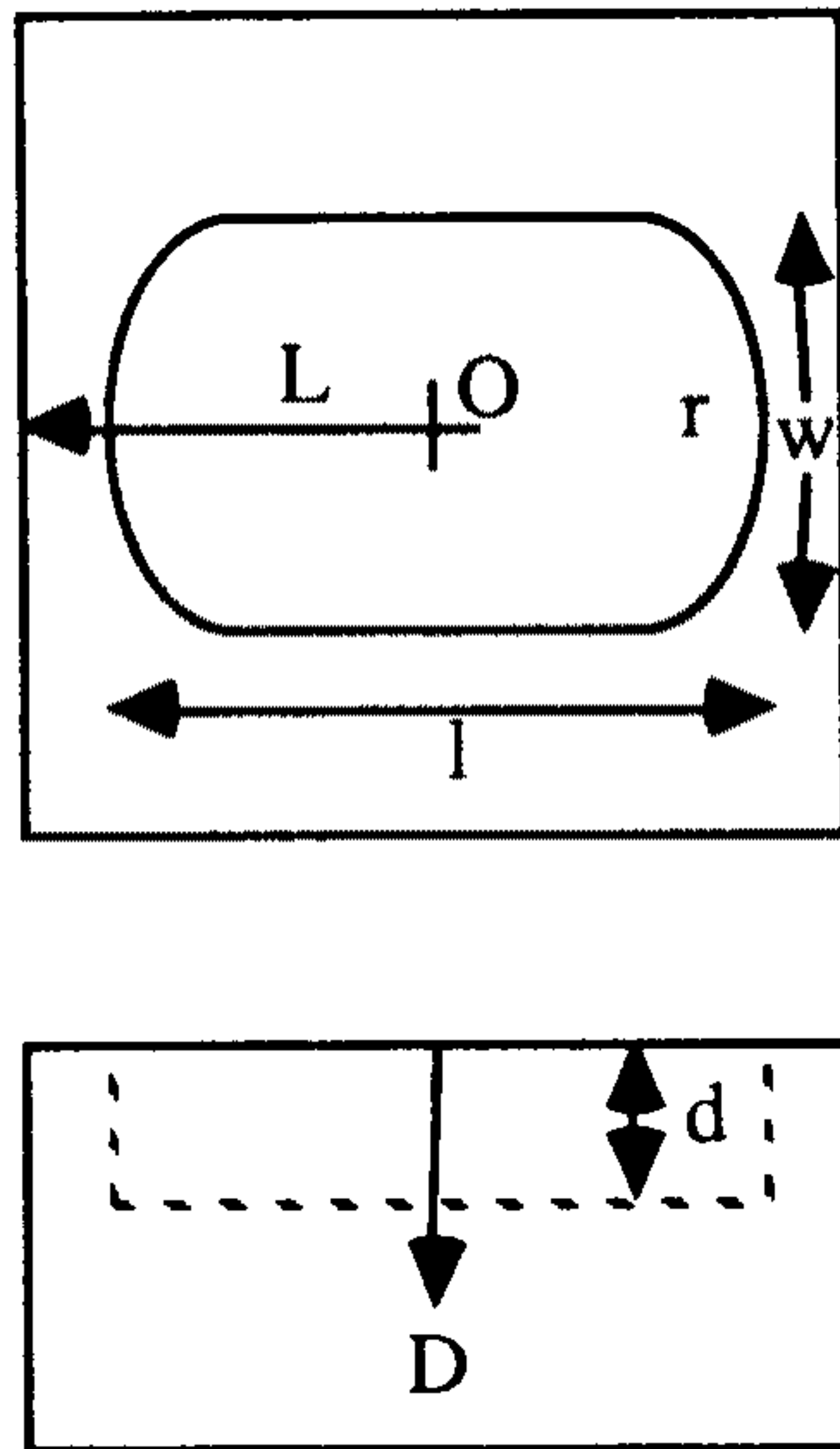


그림 2-3. Moldsys data의 COPS로의 mapping 예  
(Atomic feature:pocket\_5)

위의 그림은 atomic feature 'pocket\_5'에 대한 형상정의이고 이에 해당하는 parametric data file에는 다음과 같은 형상정보가 저장된다.

atomic feature : pocket\_5

- 1) identification no. : 일련번호.
- 2) Origin : point 0 의 position vector (x,y,z).
- 3) Direction of D : 깊이 방향의 direction vector (x,y,z).
- 4) Direction of L : 길이 방향의 direction vector (x,y,z).
- 5) width (w)
- 6) length (l)

7) depth (d)

8) radius of side (r)

작업설계의 가공feature인 pocket 밀링은 입력으로 pocket 폭, pocket 길이, 절삭할 깊이, 최소corner radius 등이 필요한데 상기 parameter로 부터 입력 데이터를 취한다.

### 제 3 절 작업 설계 메인 모듈의 구성과 기능

절삭공정 별로 서브루틴들이 만들어 졌는데, 루틴 별로 약간의 차이는 있으나, 그림 2-4의 Flow chart같은 작업들이 공통적으로 수행된다.

프로그램은 그림 2-5와 같이 구성되어 있고, 각 공정별 수행기능은 다음과 같다.

#### 1. 밀링공정 - 면삭가공, 측면밀링, 슬롯 및 포켓밀링

##### 가. 공구선택 및 절삭깊이 계산

Face밀링의 경우에는 절삭폭의 1.5배 이상인 face mill 중에서 최소의 것을 선택하거나, 해당 공구 (공구직경이 face밀 폭의 1.5배 이상)가 없는 경우, 가용 공구 중 최대 직경을 선택한다. Side milling, pocket milling 등은 Atomic feature의 형상 제한에 따라 공구경을 결정하고, 절삭조건 file에서 최대허용 절삭깊이를 검색하여 결정한다.

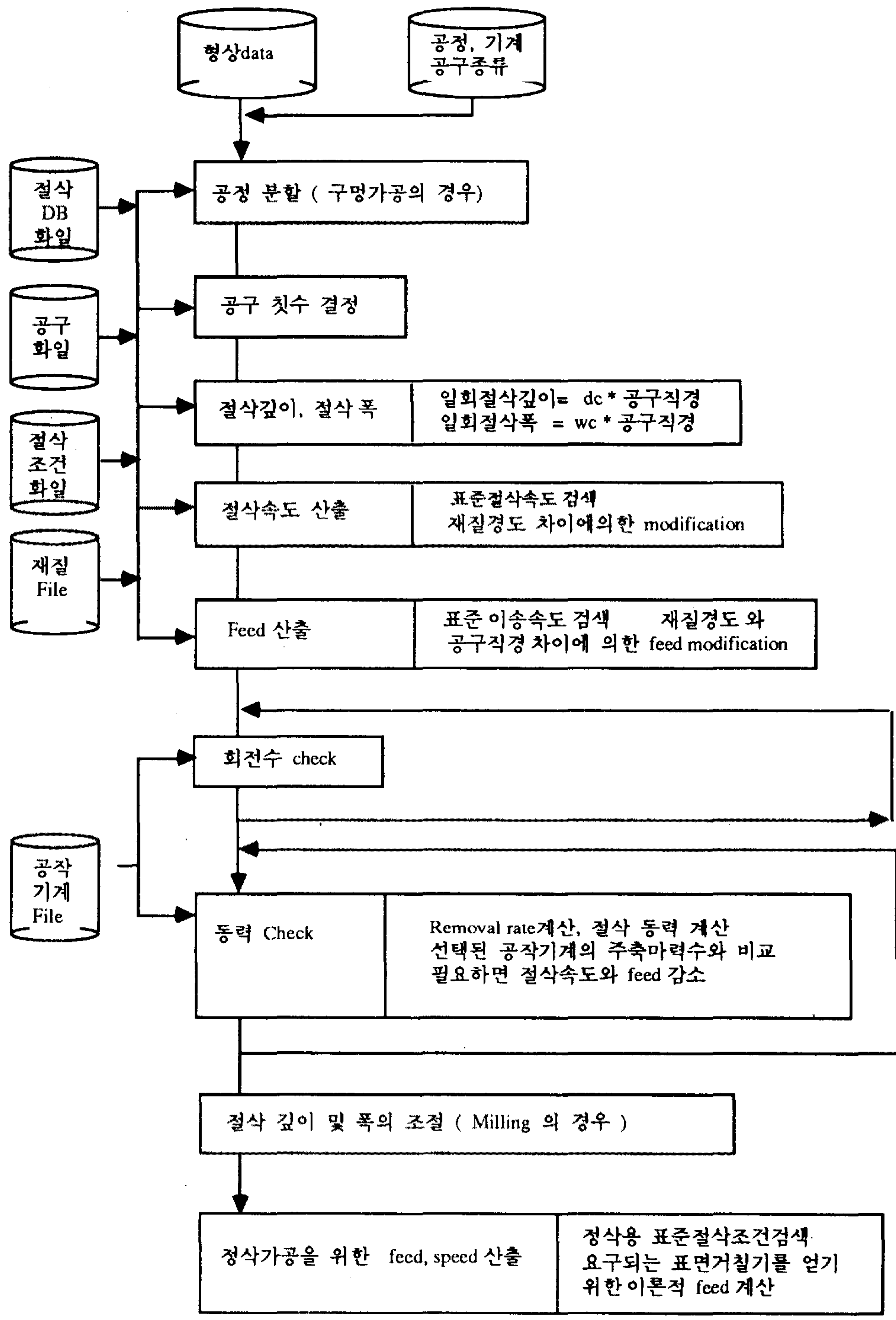


그림 2-4. 절삭 parameter 산출 기능의 Flow diagram

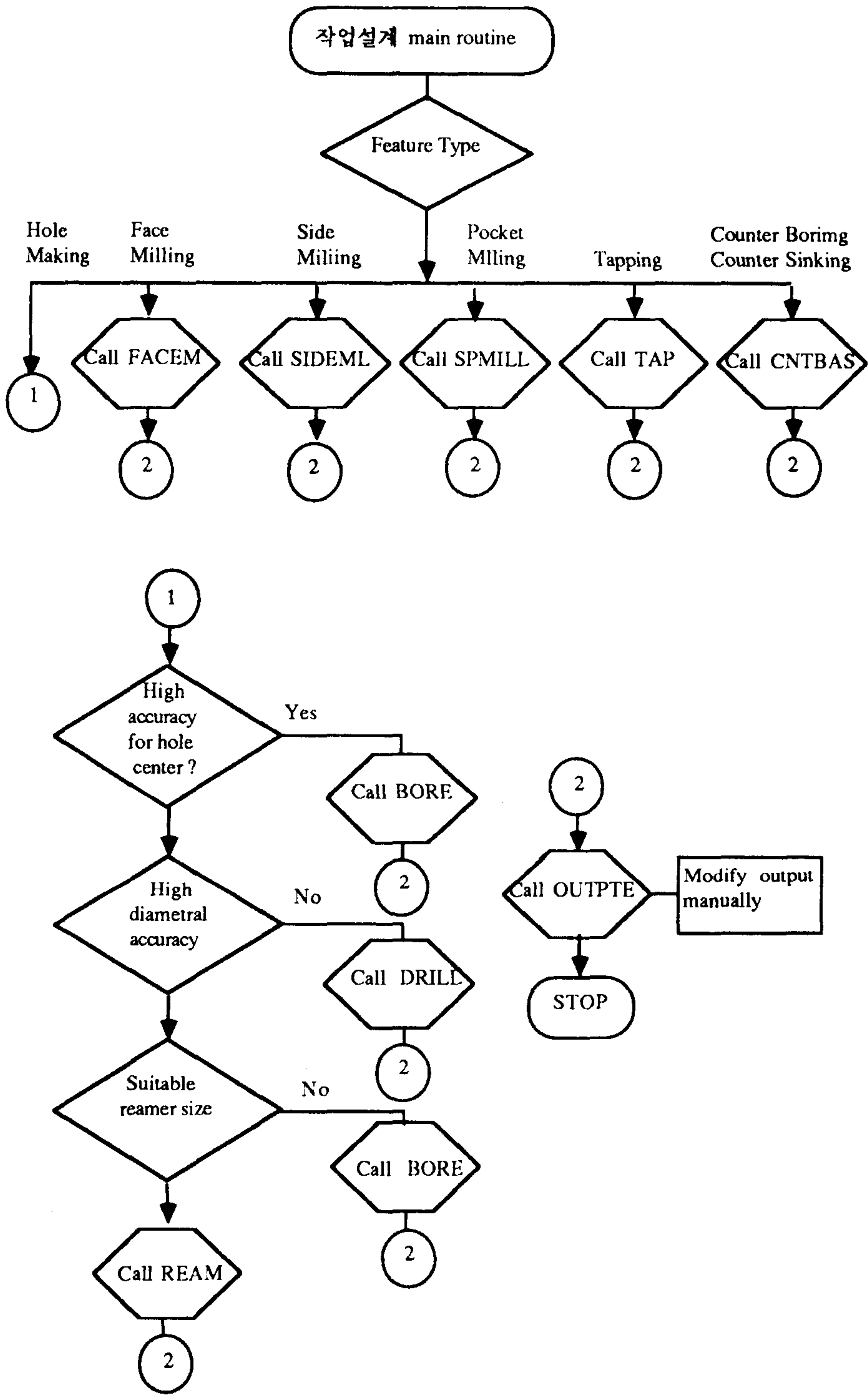


그림 2-5. 절삭조건 산출 모듈의 프로그램 구성

나. 절삭속도 및 feedrate

다음장에서 기술되는 절삭성 데이터베이스에서 절삭계수들을 검색하고 이를 절삭조건결정 subroutine에 대입하여 해당 절삭속도와 feedrate를 계산한 다음 이를 다시 feature별 modify rule에 의해 갱신한다.

다. 공작기계용량 check

선정된 절삭속도로부터 공구 회전수를 계산하고, 공작기계 file로부터 실제 사용할 공작기계의 최대 주축회전수를 검색하여, 두값을 비교, 허용치 안에 들지 않으면 허용 회전속도를 기준으로 절삭속도를 다시 절감한다. 절삭속도 및 feed, 재질 경도를 입력으로 소요절삭동력을 계산하여, 사용할 기계의 주축 마력수와 비교, 허용치보다 크면, 속도와 feed를 10%씩 감소시키며, 허용치 안에 들 때까지 이를 반복한다.

일반적으로 절삭가공에 소요되는 동력은 다음식과 같이 나타낼 수 있다.

$$P_m = P_s * Z_w$$

$P_s$  = 단위 체적당 소요 가공 에너지

$Z_w$  = metal removal rate

[Ref. Boothroyd "Fundamentals of Metal Machining and Machine Tools"]에 의하면  $P_s$ 는 재질, 절삭조건에 따라 다음식들로 나타낼 수 있다.

$$P_s = \log^{-1}( a_1 * \log A + a_2 )$$

$a_1 = -0.468$	$a_2 = 9.14$	탄소강
$-0.478$	9.28	합금강
$-0.457$	8.97	주철

A : mean undeformed chip thickness

여기서 mean undeformed chip thickness는 하나의 공구날에 의해 깎여 생성되는, 이론적 chip 두께로, 공정에 따라 표 2-1의 식들로 표현된다.

단위시간당 절삭되는 양인 metal removal rate도 표 2-1에 수록되어 있다.

표 2-1. 공정별 비절삭량 및 Mean Undeformed Chip 두께 산출식

Operation	Metal Removal Rate	Mean undeformed chip thickness
Boring	$\pi f a_p n_w (d_M - a_p)$	$\sin k_r$
Drilling	$\frac{\pi f d_m^2 n_t}{4}$	$\frac{f}{2} \sin k_r$
Face Milling	$a_e a_p V_f$	$\frac{V_f}{N_{nt}}$
Peripheral Milling		$\frac{V_f}{N_{nt}} \sqrt{\frac{a_e}{d_t}}$

$a_f$  = 절삭폭

$a_p$  = 절삭깊이

$d_m$  = 가공부직경

$d_t$  = 공구경

$n_t$  = 공구회전수

$N$  = 절삭날 수

$V_f$  = 이송속도

$k_r$  = 주절삭각

라. 절삭 pass수 계산

정삭이 필요한 경우 0.5mm를 남겨두고 절삭하여야 할 폭과 깊이를 아래와 같이 분할하여 pass수효를 계산 한다.

$$n_T = n_1 \cdot n_2$$

$$n_1 = \text{INT} (W/w+0.99)$$

$$n_2 = \text{INT} (H/d+0.99)$$

W = milling 할 폭 (mm)

w = 공구경의 2/3 혹은 공구경(폭의 1.5배 이하 공구)  
+ 면삭길이(면삭) = 절삭폭 (측면밀링, 포켓밀링 등)

H = 절삭할 깊이

D = 최대 허용절삭깊이

n<sub>1</sub> = 절삭할 면적을 cover하는데 소요 되는 절삭횟수

n<sub>2</sub> = 절삭할 깊이를 cover하는데 소요 되는 절삭횟수

그리고 절삭깊이 d는 모든 pass에서 동일하도록 다음과 같이 조절한다.

$$d = H/n_2$$

마. 정삭 data

절삭조건화일로부터 검색된 절삭계수와 절삭조건 결정프로그램을 통해 계산된 표준 값을, 요구되는 표면거칠기를 생성하기 위해 필요한 feed의 계산값과 비교하여, 두값 중 작은 값을 정삭feed로 선택한다. 요구되는 표면 거칠기를 얻기 위한 face밀링 정삭feed 계산식은 cutter mark의 기하학적 관계로부터 도출되며, 아래와 같다.



a. Single point tool을 사용하는 경우

$$f = 4\alpha h (\cot k_x + \cot k'_x)$$

f : feed(mm/rpm)

h : surface roughness value(mm)

$k_x$  : major cutting edge angle

$k'_x$  : minor cutting edge angle

$\alpha$  : 2.3 (safety factor)

b. Tool nose radius가 주어진 경우

$$f = \alpha \sqrt{8hr}$$

f : feed (mm/rev.)

h : surface roughness value (mm)

r : Tool nose radius (mm)

$\alpha$  : 2.3 (safty factor)

## 2. 구멍 가공

구멍의 크기, 중심 위치정밀도, 직경치수, 정밀도에 따라 아래와 같이 공정이 결정된다.

- 1) 드릴링 한도 (Cutting Technologe File에 저장되어 있음)보다 직경이 크면 보링
- 2) 위치 정밀도가 요구되면 보링
- 3) 칫수 정밀도가 요구되면 보링
- 4) 표면조도가 요구되고 적합한 크기의 리머가 있으면 reaming,  
아니면 보링
- 5) 상기 해당사항이 없으면 드릴링

## 가. 드릴링 공정

드릴작업에 의해 구멍 가공을 할 경우 센터드릴, 파일럿트 드릴, 일차 황삭, 이차 황삭, 정식드릴 등 연속적인 드릴링 공정에 의해 최종치수를 얻게 된다.

드릴링 스텝을 결정하기 위한 logic은 다음과 같다. 각 변수들이

- D = 구멍의 직경
- D<sub>P</sub> = 파일럿트 드릴링 직경
- D<sub>R1</sub> = 일차 황삭 직경
- D<sub>R2</sub> = 이차 황삭 직경
- C<sub>1</sub> = 드릴링 최소 직경
- C<sub>2</sub> = 파일럿트 드릴링 최소한도
- C<sub>3</sub> = 일차 황삭 최소한도
- C<sub>4</sub> = 이차 황삭 최소한도
- C<sub>5</sub> = 드릴링 최대 직경
- C<sub>6</sub> = 파일럿트 드릴링 직경 factor
- C<sub>7</sub> = C 보다 큰 구멍을 가공할때 파일럿트 드릴링 직경 factor
- C<sub>8</sub> = C 보다 크고 C 이하의 구멍 황삭 직경 factor
- C<sub>9</sub> = C 보다 큰 구멍 황삭 직경 factor
- C<sub>10</sub> = 이차 황삭 직경 factor

일 때

I. if  $C_1 < D < 3$

직접 드릴

II. if  $3 < D < C_2$

1)  $D_p$  로 파일럿 드릴링 후

$$D_p = C_6 D$$

2) 최종 칫수 드릴

III. if  $C_2 < D < C_3$

1) 센터 드릴링

2)  $D_p$  로 파일럿 드릴링

$$D_p = C_7 D$$

3) 최종 칫수 드릴

IV. if  $C_3 < D < C_4$

1) 센터 드릴링

2)  $D_p$  로 파일럿 드릴링

$$D_p = C_7 D$$

3)  $D_{R1}$  로 황삭

$$D_{R1} = C_8 D$$

4) 최종 칫수 드릴

V. if  $C_4 < D < C_5$

1) 센터 드릴링

2)  $D_p$  로 파일럿 드릴링

$$D_p = C_7 D$$

3)  $D_{R1}$  로 황삭

$$D_{R1} = C_9 D$$

4) 이차황삭 드릴링

$$D_{R2} = C_{10} D$$

5) 최종 칫수 드릴

C 값은, 제 2장 3절에 설명된, 절삭가공 data base file ( Cutting Technology File ) 에 지정되어 있어, 저장된 data를 바꿔 줌으로서 작업장 여건이나 policy에 맞게 상기의 다섯 단계를 조절할 수 있다. ( 예 : C<sub>4</sub> 를 C<sub>5</sub> 값과 같게 설정하면 이차 황삭 단계가 없어짐 )

CTF file에 저장된 data를 사용하여 센터드릴 직경 깊이등을 결정한다.

절삭조건 결정프로그램에서 표준 speed, feed 값을 결정한 후 milling에서와 유사한 방법으로, rpm 및 power를 check하고 필요하면 이를 modify한다.

결정된 speed, feed값을 구멍의 깊이에 따라, 아래와 같이 modify 한다.

- 1) if depth / diameter > 3  
    speed x 0.9    feed x 0.9
- 2) if depth / diameter > 4  
    speed x 0.8    feed x 0.9
- 3) if depth / diameter > 5  
    speed x 0.7    feed x 0.8
- 4) if depth / diameter > 6  
    speed x 0.75    feed x 0.8
- 5) if depth / diameter > 7  
    Gun drill 사용

#### 나. 보링 공정

보링은 드릴링 후 사상공정으로, 이때의 황삭 드릴링 직경은 아래와 같이 결정한다.

$$D_R = D - 2 * DC - C_f$$

여기서  $D$  = 요구 직경  
 $D_R$  = Drill 직경  
 $DC$  = Boring의 허용절삭 깊이 ( 절삭조건 file로 부터 )  
 $C_f$  = 정삭의 여유

여기서도 절삭조건 결정프로그램에서 표준 speed, feed 값을 결정한 후 밀링공정에서와 유사한 방법으로 rpm 및 power check, modify 과정을 수행한다.

#### 다. 리밍

구멍사상작업인 reaming의 rologic은 다음과 같다.

1) if  $C_1 < D < C_2$

$$D_R = D - C_5$$

2) if  $C_2 < D < C_3$

$$D_R = D - C_6$$

3) if  $C_3 < D < C_4$

$$D_R = D - C_7$$

여기서  $D$  = 최종구멍 치수  
 $D_R$  = 황삭 드릴링 사이즈  
 $C_1$  = 리밍 최소 직경  
 $C_2$  = 경우 1)의 상한 직경  
 $C_3$  = 경우 2)의 상한 직경  
 $C_4$  = 리밍 최대 직경  
 $C_5$  = 경우 1)의 reaming 여유  
 $C_6$  = 경우 2)의 reaming 여유  
 $C_7$  = 경우 3)의 reaming 여유

C 값은 CTF file에 저장되어 있어, 이 값을 조정함으로써 구멍크기에 따라 reaming 여유를 설정할 수 있다.

황삭은 drilling routine에서 절삭 parameter를 결정하고, reaming을 위한 feed, speed를 절삭조건 결정프로그램으로부터 계산된 후, rpm 및 power check, modify 과정을 수행한다.

### 3. 나사가공

나사는 호칭치수와 pitch, 나사 깊이로 정의되고 호칭치수에 따른 드릴링 사이즈 결정이 다음과 같이 이루어진다.

$$D_R = D - C_5 P$$

$D_R$  = 드릴링 직경

$D$  = 나사호칭치수

$P$  = pitch

$C_5$  = 드릴링 직경 factor

드릴링공정은 드릴링 routine에 의해 절삭조건이 결정되고, threading속도는 절삭조건 결정프로그램에 의해 결정된다.

### 4. Counterboring / Countersinking

나사가공과 유사한 방법으로 황삭 드릴링 사이즈를 결정하고, Feed, 속도를 절삭조건 결정프로그램에 의해 결정된다.

## 제 4 절 지원 데이터 베이스 file

### 1. 금형용 재질 file

금형부품용 재료의 절삭가공에 중점을 두기 위하여 국내에서 생산 및 구입가능한 금형재료를 망라하여 재질 file을 구성하여 보다 구체적인 재료 선택을 할 수 있도록 하였다. 상품명으로 선택된 재질은 프로그램내에서 AISI기호로 바뀌어 절삭조건결정시 사용된다.

### 2. 공작기계 file

공작기계 file에는 가용공작기계의 주축속도, 마력수, 각축 최대이송거리 등의 공작기계 용량이 저장되게 되어 있어, 가공 파라미터 선정에 참조가 된다.

### 3. 절삭공구 file

가용공구의 코드, 공구재질, 가용공구직경, 주절삭각, 부절삭각, tool nose radius 등이 수록되어 있다.

### 4. 절삭조건 file

12개의 공구 category로 구분되어 피삭재의 재질과 절삭할 직경에 따라 절삭깊이, 절삭폭, 황삭/정삭 주축속도, 황삭/정삭 이송속도가 수록되어 있던 1차년도 Handbook type의 절삭 정보를 개선하여, 가공제어인자들 사이의 매핑관계를 공리화하여 신경회로망의 숨은층(hidden layer)으로 구



성한 절삭가공조건의 산출과정을 프로그래밍하였고, 이를 통하여 요구되는 절삭상태가 얻어지도록 절삭조건이 결정되는 subroutine을 추가하였다. 그리고 이 subroutine에 입력되는 절삭계수들은, 수집된 전문가의 knowhow를 Fuzzy값 처리하는 DB editor에 의해서 절삭계수 데이터베이스로 작성되어 구성되도록 하였다.

#### 5. 절삭기술 화일 (Cutting Technology File)

Cutting Technology File에는, 황삭후 사상이 필요한 표면거칠기, drill작업의 최대직경, reaming작업이 가능한 최소직경 등 작업장 고유의 가공원칙 또는 가공 knowhow를 테이블 형태로 수록하고 있다. 작업설계 프로그램에서는 이들을 변수로 사용하기 때문에, 프로그램과는 별도로 이 데이터베이스의 생성, 변경, 보완 등이 가능하다. 아래의 표 2-2에는 Cutting Technology File의 내부 중 일부가 보여지고 있다.

#### 6. Data file 보수유지 프로그램

상기 데이터화일 내용은 작업설계 시스템을 사용할 환경 (작업장)마다 상이하고, 한번 작성된 file을 환경의 변화 (예; 새로운 기계의 도입) 에 따라 수시로 변경할 수 있어야 하기 때문에 이 목적을 위하여 file maintenance program을 개발하였다.

- 1) File Creation : 새로운 화일 생성
- 2) File Editing : 기존화일 재구성
- 3) File Print out : 읽기 편한 format으로 프린트하기 위하여  
data file 생성
- 4) File Listing on monitor : 모니터상에 화일 내용 디스플레이

의 기능을 가지고 있고, 모든작업이 시스템과 대화 형식으로 이루어 진다. 재질, 공작기계, 공구 화일도 이프로그램에 의하여 관리하게 된다. 그러나 cutting condition file들을 별도의 절삭조건 DB editor에 의해 관리된다.

표 2-2 절삭 기술 file

드릴링 Parameter	Code	Location
최소 드릴링 직경	DR1	Z(1)
최대 드릴링 직경	DR2	Z(5)
황삭 드릴이 필요한 최소직경	DR3	Z(3)
∴	∴	∴
밀링 Parameter		
사상이 요구되는 표면거칠기	ML1	Z(31)
사상 feed factor	ML2	Z(32)
사상 speed factor	ML3	Z(33)
∴	∴	∴
Tapping Parameter		
나사구멍 드릴링 factor	TP1	Z(14)
탄소강 가공시 최소 나사길이 factor	TP2	Z(15)
∴	∴	∴

## 제 5 절 입출력 양식

제 2절에서 언급된 바와 같이 MOLDSYS에서 설계정보를 5개 파일로 받아들이고 MOLDCAPP으로부터 공정정보를 별도의 textfile로 받아 들인다.

Interface 프로그램에 의해, 공정에 관련된 설계정보를 수록한 파일과, 정보의 위치를 검색하여 필요한 입력 파라미터들을 작업설계 모듈에 공급하게 된다. 그림 2-6은 전화기금형의 설계정보를 수록한 데이터파일들의 구조와 상호관계를 보여주고 있다.

COPS의 출력은 공작기계 코드, 공구종류, 공구크기, 절삭조건 등 각 절삭 공정에서 필요한 모든 파라미터를 포함하고 있고, MOLDCAPP의 공수계산 모듈에서 산출된 표준시간을 포함하여 최종적으로 작업지시서의 형태로 출력한다. 작업설계의 결과를 검토하기 위하여 COPS출력 양식을 별도로 마련하였다. 전화기금형 세트 중 한 부품인 분리형 캐비티 플레이트 (DCAP) ( 그림 2-7 )을 가공하기 위한 작업설계 결과의 일부는 그림 2-8과 같다. 기능형상 별로 공정이 전개되고 각 기능형상에 소속된 기본형상들이 개별 절삭 공정의 단위로서 출력된다.

SET file ..

MOLD.SET	
Set_id	Mold name
1	TELEPHON
2	UPPRCASE

PART file ..

TELEPHON.PRT						
Set_id	Part_id	Part name	Plate Width	Plate Length	Plate Height	Material
1	1	BTCP	250.0	250.0	60.0	S55
1	2	DCAP	250.0	230.0	70.0	S55C
1	3	CACO	150.0	180.0	30.0	S55C

Functional Feature file ...

BTCP.FNC			
F.F_id	Process name	Part_id	Parent_id
1	raw-material	1	0
2	general-hole	1	1
3	sprue-bush-hole	1	1

Atomic Feature file ...

BTCP.ATM			
A.F_id	Atomic Feature name	F.F_id	Parent_id
1	surf-1	1	0
2	surf-1	1	1
3	hole-3	2	0

Atomic Feature의 Parameter file ...

BTCP.PRM						
A.F_id	Origin_x	Origin_y	Origin_z	direction vector_x	direction vector_y	----
1	125	0	30	1	-1	----
2	125	125	0	-1	1	----
3	30	25	70	0	0	----
						----

그림 2-6. 설계정보 파일 구조

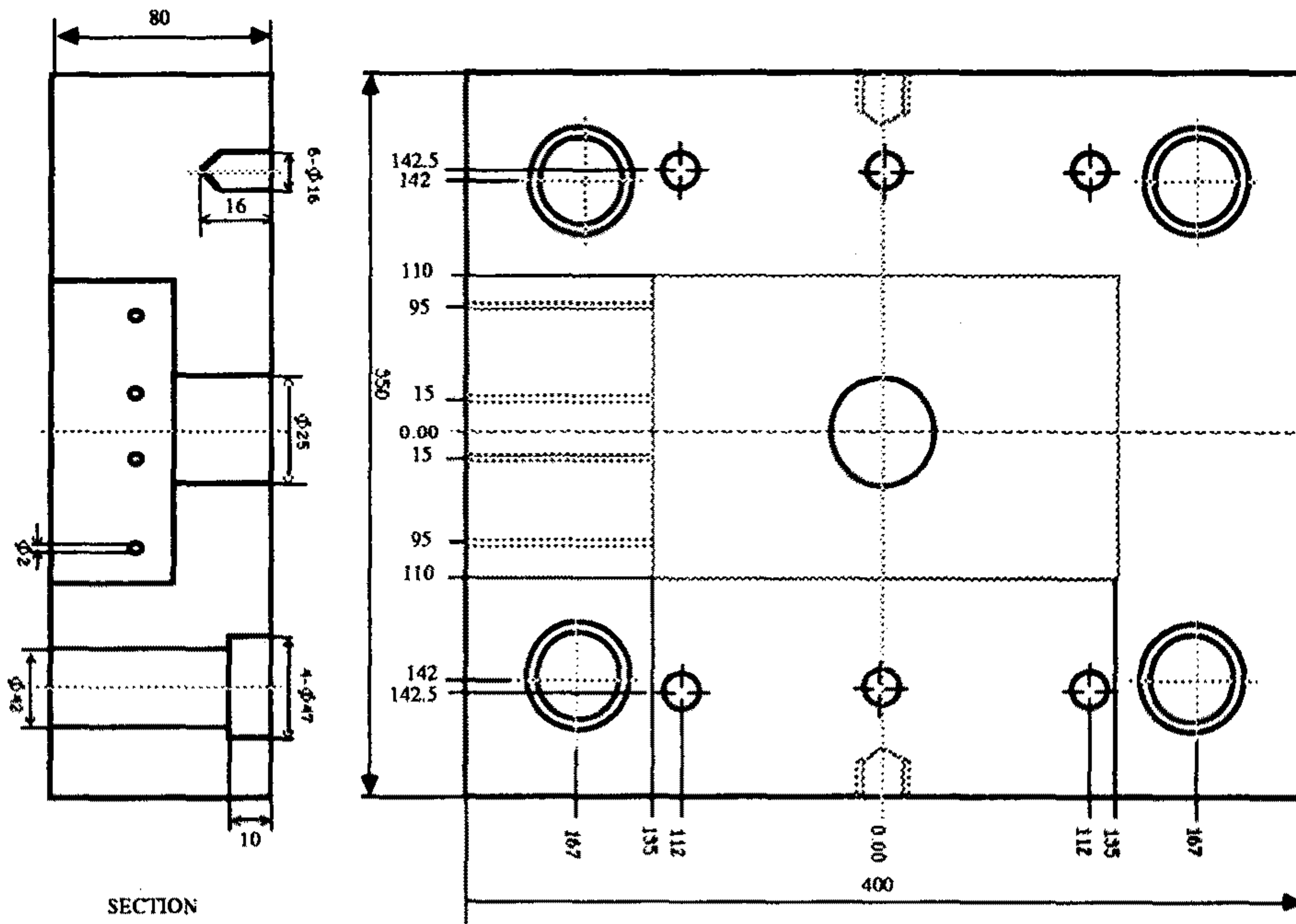


그림 2-7. 분리형 캐비티 플레이트 (DCAP) 도면

===== ROUTING SHEET =====

=> Set # 11 .. telephon

====> Part # 40 .. dcap

=====> PROCESS # 1 .. rough-cutting                      feed unit:(mm/min for milling, mm/rev for others)

Oper. num.	Atomic name	Ftr. num.	Oper. Descrp.	Tool type	dia.	Machine Tool type	name	Pass	Feed	Speed (m/sec)	RPM	Depth (mm)	Width (mm)
1	surf-1	1	11 rough cut	2	500.0	MF	IVERT. MILL	2	332	1.00	38	4.8	0.0
2	surf-1	1	11 finish cut	2	500.0	MF	IVERT. MILL	1	110	1.26	47	0.5	0.0
3	surf-1	2	11 rough cut	2	500.0	MF	IVERT. MILL	2	332	1.00	38	4.8	0.0
4	surf-1	2	11 finish cut	2	500.0	MF	IVERT. MILL	1	110	1.26	47	0.5	0.0
5	surf-1	3	11 rough cut	2	500.0	MF	IVERT. MILL	2	332	1.00	38	4.8	0.0
6	surf-1	3	11 finish cut	2	500.0	MF	IVERT. MILL	1	110	1.26	47	0.5	0.0
7	surf-1	4	11 rough cut	2	500.0	MF	IVERT. MILL	2	332	1.00	38	4.8	0.0
8	surf-1	4	11 finish cut	2	500.0	MF	IVERT. MILL	1	110	1.26	47	0.5	0.0
9	surf-1	5	11 rough cut	2	125.0	MF	IVERT. MILL	2	577	2.10	320	4.8	0.0
10	surf-1	5	11 finish cut	2	125.0	MF	IVERT. MILL	1	213	2.63	401	0.5	0.0
11	surf-1	6	11 rough cut	2	125.0	MF	IVERT. MILL	2	577	2.10	320	4.8	0.0
12	surf-1	6	11 finish cut	2	125.0	MF	IVERT. MILL	1	213	2.63	401	0.5	0.0

=====> PROCESS # 2 .. square-pocket-for-insert

Oper. num.	Atomic name	Ftr. num.	Oper. Descrp.	Tool type	dia.	Machine Tool type	name	Pass	Feed	Speed (m/sec)	RPM	Depth (mm)	Width (mm)
13	pock-1	13	14 rough cut	3	50.0	MS	ITAC-V1 Mill	6	26	0.35	133	29.8	50.0
14	pock-1	13	14 finish cut	3	50.0	MS	ITAC-V1 Mill	3	33	0.44	167	0.5	50.0

=====> PROCESS # 3 .. sprue-bush-hole-shape-processing

Oper. num.	Atomic name	Ftr. num.	Oper. Descrp.	Tool type	dia.	Machine Tool type	name	Pass	Feed	Speed (m/sec)	RPM	Depth (mm)	Width (mm)
15	hole-3	14	14 rough cut	3	20.0	MS	ITAC-V1 Mill	1	83	0.35	334	19.5	20.0
16	hole-3	14	14 finish cut	3	20.0	MS	ITAC-V1 Mill	1	104	0.44	417	0.5	20.0

그림 2-8. 작업설계 결과 출력



# 제 3 장    절삭가공인자의 공리화와           매핑관계의 구성

## 제 1 절    절삭가공인자의 결정

### 1. 개요

작업설계 시스템과 같은 자동화된 공정계획 단계에서 필요로 하는 절삭성 데이터들의 성격에 비추어 볼때 기존에 산적되어 핸드북 수준의 절삭성 데이터들은 일반적으로 크게 다음과 같은 3가지 형태의 문제점들을 가지고 있다고 볼 수 있다.

(1) 표준(추천) 절삭조건표에 의해서 주어지고 있는 절삭성 데이터들은 우선 절삭 환경인자의 다변수화가 불분명하고 ( 많은 인자를 함께 고려 해야 하기 때문 ), 절삭인자가 정확하게 매칭되지 않는 경우에 그 불확실성이 매우 높으며, 실험 또는 경험에 의존하기 때문에 광범위한 데이터의 양을 확보하기가 매우 힘들다.

(2) 절삭인자에 대한 절삭상태의 예측이 가능하다고 하더라도, 그 역의 관계는 불분명한 것이 대부분이다.

(3) 가공이 실시되고 있는 상황에 있어서도 가공인자를 동적인 변수로 취급하는가 정적인 변수로 취급하는가에 따라 가공상황은 매우 큰 차이를 보이며, 가공인자의 중요도에 따라서 변수값의 크기에 따른 경향도 다르다.

한편 COPS에서 절삭가공 파라미터를 설정하는 과정은 「주어진 가공인자와 요구되는 가공상태를 입력으로 하여 가공제어인자를 출력」 하는 것이라 할 수 있는데, 이러한 성격의 데이터 처리에는 전문가에 의해 얻어진 휴리스틱, 수치 공식에 의한 알고리즘, 데이터베이스 테이블 등의 방법을 사용하여 이러한 시스템을 구현하는 것이 바람직 할 것으로 판단되었다.



이러한 방법으로 처리하게 되면 다음과 같은 장점들이 기대될 수 있을 것으로 여겨지기 때문이다.

(1) 데이터들 상호간의 경향이 추출될 수 있고, 그 경향의 입장에서 데이터가 재 정리될 수 있으므로 위험한 ( 극히 가식적이거나, 그 불규칙 데이터의 오차가 극히 큰 ) 데이터를 안정시켜 줄 수 있다.

(2) 개선된 데이터들은 또한 학습기능(learning performace)에 의해서 모형화 할 수 있는 장점이 있다.

(3) 신뢰확률이 Fuzzy 숫자의 형태로 반영될 수 있고, 여기에 modify 또는 reduction 룰의 적용이 가능하다.

(4) 핸드북, 가이드북, 공구카타로그, 경험적인 수집 데이터들의 차이점을 반영시킬 수 있다.

(5) 이들 데이터를 작업설계 단계에서 참조로 사용할 때에 실시간 이내에서 처리가 가능하고, 데이터베이스에 없는 특정한 값도 처리할 수 있다.

본 연구에서는 편의상 가공 인자, 가공 상태, 가공 제어 인자라는 용어를 다음과 같이 정의하여 사용하고자 한다.

가공 인자 ( process parameter ) : 가공을 하기 전에 미리 확정되어

MOLDSYS나 MOLDCAPP 시스템에서 전송되어 오는 기본 원형등에 관련된 정보 및 가공상황정보. 여기에는 functional feature, automic feture, 가공 feature, 공구 category , 공구의 재질과 기하학적 형상, 공구의 상태, 공작물의 재질 및 상태, 공작기계의 제원 및 상태에 관한 정보가 포함된다.

가공 상태 ( process state ) : 가공의 진행도중이나 또는 결과로 생기는

가공들의 상태를 의미한다. 여기에는 절삭저항, 절삭 동력, 표면 거칠기, 칩의 상태 등을 의미한다. 여기에는 절삭저항, 절삭 동력, 표면 거칠기, 칩의 상태, 공구 마멸, 가공 정밀도 등이 포함된다.

가공 제어 인자 ( process control parameter ) : 원하는 가공 상태를 유도하기 위하여 COPS에서 조절할 수 있는 인자, 즉 COPS의 출력을 의미한다. 일반적으로 공구 size, 절삭속도, 이송속도, 절삭 폭, 절삭 깊이를 의미하게 된다.

## 2. 가공 상태에 영향을 미치는 인자

주어진 가공인자와 요구되는 가공상태를 입력으로 하여 가공제어인자를 출력을 하는 시스템을 구성하기 위해서, 가공상태에 대하여 중요한 인과관계를 형성하는 가공인자와 가공제어인자를 파악하고 정상적인 인과관계를 살펴보면 다음과 같다.

### (1) 절삭 동력

절삭동력에 영향을 미치는 인자로서는 절삭깊이, 절삭폭, 이송속도, 절삭속도, 경사각, 어프로치각, 전단각, 피삭재의 재질(전단변형응력, 인장강도, 브리넬경도 등), 코너 반경, 칩 브레이커 구조 등이 있게 된다. 절삭동력에 큰 영향을 미치는 절삭깊이, 절삭폭, 이송, 인장강도, 경도 등은 절삭동력과 단조관계(monotonic relation)을 갖게 되며 기타 작은 영향을 미치는 다른 인자들과의 인과관계는 다음과 같다.

( 날끝 경사각 ↑ ) ⇒ ( 유출각 ↑ ) ⇒ ( 배분력 ↑, 이송분력 ↓ )

( 어프로치각 ↑ ) ⇒ ( 주분력 ↑, 배분력 ↑, 이송분력 ↓ )

( 코너 반경 ↑ ) ⇒ ( 배분력 ↑, 이송분력 ↓ )

흠형 칩 브레이커 : ( 절삭깊이 > 랜드 폭 ) ⇒

( 칩 브레이커 효과 없어짐 , 비절삭 저항 ↑ )

( 절삭 깊이 > 1/2 \* 랜드 폭 ) ⇒

( 유효 경사각 ↑, 비절삭 저항 ↓ )

( 절삭깊이 < 1/2 \* 랜드폭 ) ⇒ ( 유효 경사각 ↓, 비절삭 저항 ↑ )

## (2) 표면 거칠기

이송, 노오즈 반경, 절삭날각 등에 의해서 기하학적 거칠기가 수치공식으로 구성될 수 있다. 이러한 기하학적 요인 외의 거칠기 형성 요인은 다음과 같다.

( 절삭속도 ↑ , 경사각 ↑ ) ⇒ ( 흠에 의한 거칠기 크기 ↓ )

( 쾌삭 성분인 유황 또는 납이 첨가된 피삭재 ) ⇒

( 흠에 의한 거칠기 크기 ↑ )

( 주축 정밀도 , 이송 및 절삭깊이 변동에 의한 공구대의 안정성, 공구날끝의 동적거동 등 ) ⇒ ( 공작기계 특성에 의한 거칠기 변화 )

( 절삭속도 , 이송 ⇒ 절삭 온도 )

AND ( 가공 경화성 )

AND ( 피삭재 , 공구재질 , 절삭유 , 가공환경 ⇒ 친화성 )

⇒ ( 구성 날끝에 의한 거칠기 변화 )

( 공구재질 , 공구날부분 형상 , 피삭재 종류 , 절삭유 , 가공 분위기 ,

절삭저항 및 온도 ) ⇒ ( 공구 마멸에 의한 거칠기 변화 )

( 체터링 발생 조건 ) ⇒ ( 체터링에 의한 거칠기 변화 )

## (3) 가공 정밀도

가공정밀도로서는 치수정밀도, 형상정밀도 (진원도, 원통도, 진직도, 평면도, 면의 윤곽도, 평행도, 직각도) 등이 있게 되는데, 여기에 인과관계를 갖는 인자로서는 공작기계 요소의 위치정밀도, 설치정밀도, 이송장치의 정밀도, 열변형, 동적 변형, 운동부분의 작용력 (편심, 기어정밀도 등)에 의한 변형, 절삭력, 날끝의 위치정밀도, 형상정밀도, 공구마멸 등이 있게 된다.

#### (4) 마멸

마멸과 공구수명에 영향을 미치는 가공제어인자의 인과관계는 다음과 같다.

( 절삭속도 ↓, 이송 ↑ ) ⇒ ( 경사면 마멸 ↓ )

( 절삭깊이 ↓ ) ⇒ ( 측면 여유면 마모 ↓ )

( 이송 ↓ ) ⇒ ( 소성 변형 ↓ )

( 절삭속도 ↑ 이송 ↓ ) ⇒ ( 초기결손 칩핑 ↓ )

( 절삭속도 ↑ ) ⇒ ( 용착성 칩핑 ↓ )

#### (5) 체터링

( 이송 ↑ ) OR ( 절삭속도 ↓ )

OR ( 경사각 ↑ ) OR ( 절삭깊이 ↑ ) OR ( 노우즈 반경 ↓ )

OR ( 전방 절삭 날각 ↑ ) OR ( 측면 절삭날각 ↓ )

OR ( 절삭각 모서리 예리 ↑ ) OR ( 절삭유 사용 )

⇒ ( 체터링 발생 감소 )

#### (6) 칩의 형성

절삭 조건에 의해서 칩 브레이커의 유효범위가 정해지게 되며, 칩의 형성에 관계되는 가공제어인자는 다음과 같다.

( 이송 ↑, 절삭 속도 ↑, 경사각 ↓ ) ⇒ ( 긴 칩의 형성 ↓ )

( 이송 ↑, 절삭 속도 ↓ ) ⇒ ( 짧게 부러지는 칩의 형성 ↓ )

( 경사각, 어프로치각, 이송, 절삭깊이의 변화 ) ⇒

( 칩 배출방향의 변화 )

### 3. 매핑 관계의 구성

앞에서 고찰한 바와 같이 관련된 모든 가공인자와 가공제어인자로부터 가공상태로 매핑하는 관계는 상당히 복잡한 다단계의 상호연관 관계로 구성되어 있으며 비선형적인 관계로 구성되어 있다. 이러한 모든인자를 총괄하여 매핑관계를 구성하는 것은 지극히 어려운 일이 된다. 또한 이러한 매핑관계가 구성되더라도 주어진 가공인자와 요구되는 가공상태로부터 적절한 가공제어인자를 선정하는 것은 합성(synthesis)의 성격을 지니게 되어 더욱 어려운 문제가 된다.

본 연구에서는 일반적이라 할 수 있는 (가공인자, 가공제어 인자) ⇒ (가공 상태) 의 매핑관계 대신에 직접적인 (가공인자, 가공상태) ⇒ (가공제어인자) 의 매핑관계를 각 구성인자에 대한 세분화 및 추상화 (abstraction)를 다음과 같이 구성하였다.

#### (1) 매핑 관계의 단계별 구성

구하고자 하는 가공제어 인자 중 공구 size, 절삭속도, 이송속도, 절삭폭, 절삭깊이 모두를 하나의 매핑에 의하여 동시에 최적으로 정하는 것은 시스템의 복잡성을 증대시키게 되므로 본 연구에서는 가공제어인자를 두개의 그룹, 기하학적 인자 (공구 size, 절삭폭, 절삭깊이) 와 동적 인자 (절삭속도, 이송속도) 로 구분 하고 매핑관계를 다음과 같이 두 단계로 나누어 구성하였다.

1 단계 : (가공인자, 가공상태) ⇒ (기하학적 가공 제어 인자)

2 단계 : (가공인자, 가공상태, 기하학적 가공 제어 인자)

⇒ (동적 가공 제어 인자)

그리고 여기서 각 매핑의 입출력 관계에 대한 정의를 다음과 같이 설정할 경우를 기준으로 살펴 보기로 한다.

CLASS X1 : atomic feature의 집합 + 가공 feature

CLASS X2 : 공구 category의 집합

CLASS X3 : 공작기계의 집합

CLASS X4 : 공작물의 집합

CLASS Y1 : 절삭 동력 구속조건의 집합

CLASS Y2 : 표면 거칠기 구속 조건의 집합

CLASS Y3 : 칩의 상태 구속 조건의 집합

CLASS Y4 : 공구 마모 구속 조건의 집합

CLASS Y5 : 가공 정밀도 구속 조건의 집합

CLASS U1 : 공구의 집합

CLASS U2 : 절삭 폭의 집합

CLASS U3 : 절삭 깊이의 집합

CLASS V1 : 절삭 속도의 집합

CLASS V2 : 이송속도의 집합

가) 1 단계 매핑의 구성

1 단계 매핑은 위의 정의를 사용하여 다음과 같이 정의된다.

$$X1 \times X2 \times X3 \times X4 \times Y1 \times Y2 \times Y3 \times Y4 \times Y5$$

$$\Rightarrow U1 \times U2 \times U3$$

나) 2단계 매핑의 구성

2 단계 매핑은 위의 정의를 사용하여 다음과 같이 정의 된다.

$$X1 \times X3 \times X4 \times Y1 \times Y2 \times Y3 \times Y4 \times Y5 \times U1 \times U2 \times U3$$

$$\Rightarrow V1 \times V2$$



#### 4. 매핑 관계에서의 기본 데이터 구조

매핑 관계의 정의 및 구현에 사용되는 기본 데이터의 구조를 정의하는데 있어서 중요한 사항은 다음과 같다.

(1) 각 단계에서 사용되는 매핑관계는 독립적인 클래스를 사용하여 정의하게 된다. 데이터 구성은 위의 매핑에 관계된 클래스를 전부 포함하게 되며 매핑관계에 의하여 가공제어인자를 출력하는 과정을 클래스 기법(method)으로 정의하여 사용할 수 있다.

(2) atomic feature의 정의는 MOLDSYS와 MOLDCAPP에서 정의된 것을 그대로 사용할 수 있다.

(3) 구속 조건 :

각 구속 조건의 구분 정도(granularity)는 전문가와의 인터뷰를 통해서 결정한다. 구속 조건은 각 가공상태가 어떠한 등급에 소속되어야 하는 것을 규정하게 되는데, 소속되는 등급은 여러개로 할 수 있다.

또한 퍼지 숫자나 심볼을 사용하여 등급을 표시하는 방법도 사용할 수 있는데 이것은 전문가가 제공하는 기준에 의하여 결정한다.

각 구속조건에는 구속조건 사이의 상호관계를 정의하여 구속조건 사이의 일관성을 유지하도록 한다. 예를들어 표면거칠기의 구속조건이 어느 정도라면, 절삭동력은 반드시 어느 범위에 있어야 한다는 등의 규칙이다. 이 상호관계는 전문가가 특별히 규정한 내용에 한하여 정의한다. 또한 각 구속조건에는 구속조건의 값에 따라 이송속도와 절삭 속도를 보정하는 수정인자가 포함된다.

각 구속 조건의 등급은 대략 다음과 같이 정의 한다.

##### ① 절삭 동력 구속 조건

[MIN, MAX]의 구간을 구분 정도에 따라 n개의 SECTION 으로 분할한다.



② 표면 거칠기 구속 조건

CLASS-1, CLASS-2, CLASS-3, . . . . ., CLASS-N으로 분할한다.

③ 칩의 상태 구속 조건

CHIP MERIT MARK의 영역을 분할하여 사용하거나 칩의 특성 클래스를 구성한다.

④ 공구 마모 구속 조건

마모의 [MIN, MAX]의 구간을 구간정도의 따라 N개의 SECTION으로 분할한다.

⑤ 가공 정밀도 구속 조건

가공 정밀도의 특성 클래스를 구성한다.

(4) 1 단계 매핑

전문가와의 인터뷰를 통해 얻어진 공구 size 결정 및 공구 결정 알고리즘을 사용하여 U1의 공구를 결정한다. 또한 전문가를 통해 얻어진 다음과 같은 데이터베이스 테이블을 사용하여 표준 절삭폭과 깊이를 구한다.

복합 인덱스		표준 절삭 폭, 깊이
가공 FEATURE	피삭재 재질	
면삭 가공	S55C	5

여기서 얻어진 표준 절삭폭과 절삭깊이에 대하여 X3의 공작기계의 절삭폭 및 절삭깊이 수정인자 (correction factor)를 사용하여 보정한다.

(5) 2 단계 매핑

먼저 U1의 공구, X3의 공작기계, X4의 공작물을 사용하여 Y1, Y2, Y3, Y4, Y5를 수정한다. 그리고 표준 이송속도와 절삭속도는 전문가가 제공하는

데이터에 따라 다음의 두가지 방식중에서 선택한다.

① 전문가를 통해 얻어진 다음과 같은 데이터베이스 테이블을 사용하여 표준 이송속도와 절삭속도를 구한다.

복합 인덱스					표준 이송속도 절삭속도
가공 FEATURE	절삭깊이	절삭폭	피삭재질	공구재질	??????
면삭 가공	.....	....	S55C	.....	??????

② 다음과 같은 비선형 함수관계를 사용한다. 여기서 사용되는 비선형관계 모델은 다음 2절에서 제시된 모델중에서 선정하며, 학습데이터는 전문가로부터 얻는다.

$$\begin{aligned} \text{표준 이송속도} &= \text{function}_1 \text{ of ( 가공 feature type, 절삭깊이, } \\ &\quad \text{절삭폭, 피삭재질 type, 공구재질 type )} \\ \text{표준 절삭속도} &= \text{function}_2 \text{ of ( 가공 feature type, 절삭깊이, } \\ &\quad \text{절삭폭, 피삭재질 type, 공구재질 type )} \end{aligned}$$

#### (6) Modify Rule 의 적용

여기서 얻어진 표준이송과 절삭속도에 대하여 Y1, Y2, Y3, Y4, Y5의 수정인자를 사용하여 보정한다.

가) 공작기계 데이터 구조에는 다음의 항이 추가된다.

( 표준 절삭폭과 절삭깊이에 대한 수정인자 )

( Y1, Y2, Y3, Y4, Y5 구속조건 값의 수정인자 )

나) 공구 데이터 구조에는 다음의 항이 추가된다.

( Y1, Y2, Y3, Y4, Y5 구속조건 값의 수정인자 )

다) 피삭재질 데이터 구조에는 다음의 항이 추가된다.

( Y1, Y2, Y3, Y4, Y5 구속조건 값의 수정인자 )

#### (7) 매핑 관계의 프로그래밍

1단계 매핑은 전문가와의 인터뷰 결과에 따라 순차적인 알고리즘으로의 구성이 쉽게 가능하면 이러한 알고리즘으로 구성하고, 이러한 알고리즘이 쉽게 도출되지 않는 경우에는 BOTTOM UP 방식에 의한 규칙 베이스를 사용하였다.

매핑관계는 객체지향 프로그래밍 원칙을 사용하여 C++로 작성하는 것을 기본으로 하였다. C 언어를 사용하여 매핑관계 프레임, atomic feature 프레임, 도구 프레임, 피삭재질 프레임, 구속조건 프레임을 정의하여 사용하였다.

데이터 베이스의 구성에 있어서는 전문가와의 인터뷰에서 결정되는 데이터량에 따라 순차 파일 액세스와 메모리상의 테이블 구성을 사용하는 방식 또는 관계형 데이터베이스 (RDBMS)를 사용하는 방식중에 선택하였다. 관계형 데이터 베이스를 사용할 경우에는 C-ISAM를 이용한 INFORMIX DBMS를 사용하였으며, 다른 C언어 모듈과의 인터페이스는 INFORMIX ESQ/C를 사용하여 작성하였다.

규칙 베이스도 또한 C 언어를 사용하여 구성하였으며, 추론 방식으로는 TOP DOWN 방식과 BOTTOM UP 방식을 필요에 따라 구별하여 사용하였다. 규칙 베이스를 구성하는 베타 규칙들도 필요성이 인정되는 경우에 응용되었으며, 기타의 비단조 추론 (nonmonotonic reasoning)과 ATMS 등의 TMS구성은 필요치 않다고 판단되었다.

비선형 함수 관계는 다음 2절에서 고찰되는 방법들 중 가장 정확히 모델링되는 다층 퍼셉트론 방법을 사용하였으며, 이 모델의 구성과 학습에도 C언어가 사용되었다.

## 제 2절 매핑 관계의 구성 방법론

### 1. 개 요

앞 절에서 제시된 바와 같이 전문가가 제공하는 학습데이터를 가지고 비선형함수 관계를 구성하는 데에는, 기존의 여러가지 모델링 원칙 중 ( 가공인자, 가공상태 )  $\Rightarrow$  ( 가공제어인자 ) 의 비선형 매핑관계를 가장 잘 표현할 수 있는 방식의 선정이 필요하게 된다. 따라서 본 연구에서는 전문가가 제공하는 학습데이터에 관하여 가장 간단한 회귀 모델로 부터 GMDH 방법에 의한 모델과 신경회로망에 의한 모델링을 수행하여 가장 적절한 모델링 방법을 선정하는 과정을 수치 모델러를 통해 비교 검토하였다.

이때 기준은 예측정확도, 수렴성, 탈local minimum 등의 특성을 기존의 모형적인 프로그램을 통하여 테스트 해 보았는데, 회귀모델에 있어서는 class된 경우의 수가 너무 많아서 polynomial의 차수를 실용적인 범위에서 아무리 늘려보아도 regression 되는 그 정확도가 매우 불량하게 나왔다. 그 때문에 이 모델은 본 연구의 성격에는 적합하지 않은 것으로 판단되었고, 그 결과로 GMDH 데이터 그룹 방법에 있어서도 그 시도를 포기할 수 밖에 없었다.

그러나 다층 퍼셉트론 방법에 있어서는 다행히도 숨은층에 있어서의 수렴 과정이 비교적 잘 수행되었으며 절삭계수의 수학적 모형화가 잘 이루어짐으로써 절삭정보를 데이터베이스화 하는데에 메모리의 부족현상이 생기지 않을 것으로 기대되었으며, 임의 절삭계수를 연속적으로 선정할 수 있기 때문에 이산화된 절삭계수가 아닌 가공조건에 맞는 최적의 절삭계수를 얻을 수 있었고, 최적화 과정에 있어서도 local minimum 현상이 거의 생기지 않았다.

그리고 신경회로망 기법은 많은 양의 데이터를 병렬로 처리할 수 있고,

수학적 모형화의 수행시, 비선형성이 크고 알고리즘의 적용이 곤란한 문제를 연속적인 다변수 함수의 모형화에 응용함으로써 그 방법이 가지고 있는 학습기능에 의해서 모델링 할 수 있다는 장점을 가지고 있기도 하다.

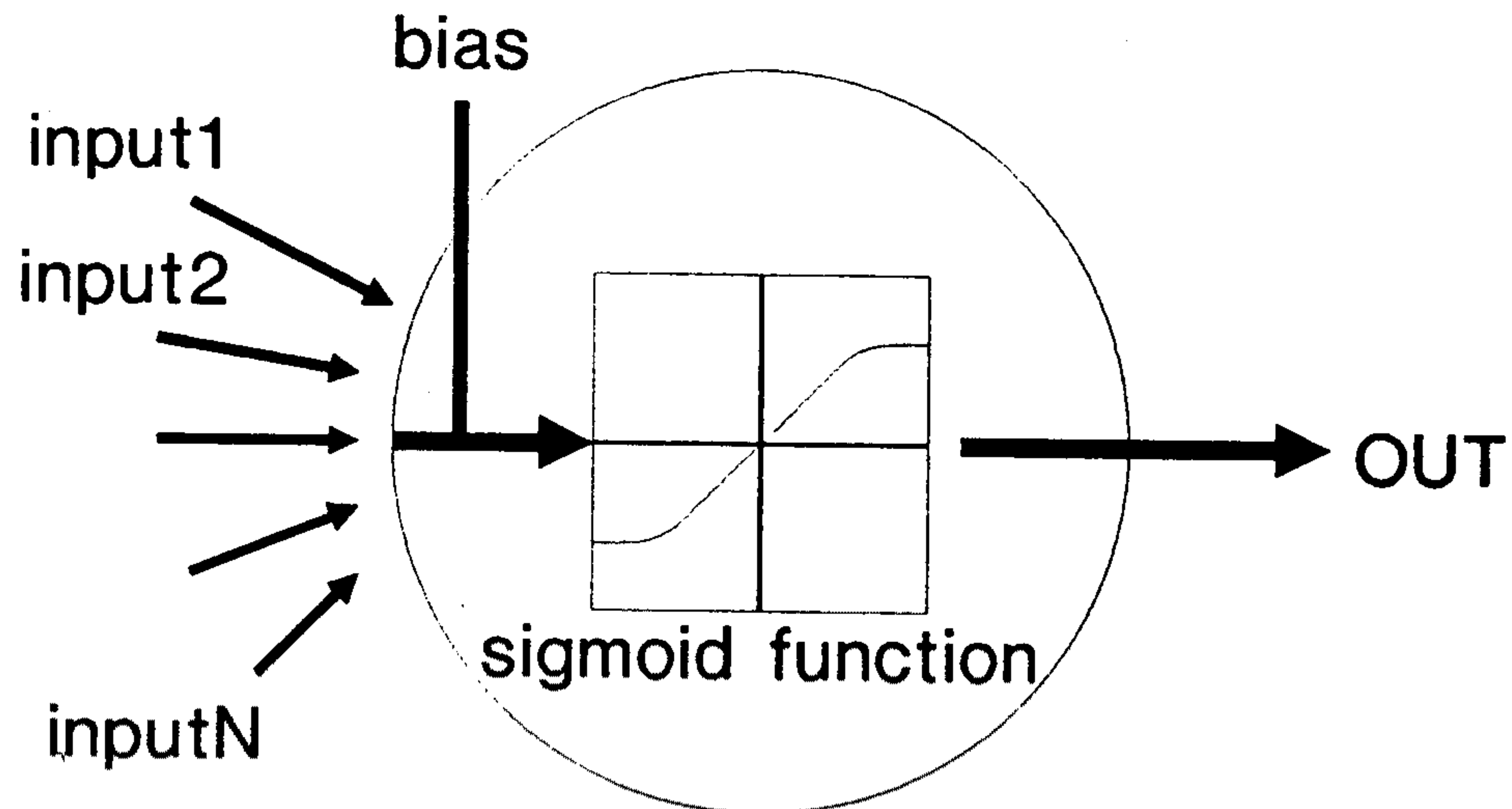
최근에는 이러한 신경회로망 기법이 스피들동력과 절삭력, 절삭온도 및 공구마모 등의 예측에 관한 응용에 있어서도 효과가 있음이 발표되고 있다.

또한 전문가가 제공하는 학습 데이터의 불확실성 처리는 삼각 퍼지숫자 또는 L-R type의 퍼지 숫자를 사용하여 처리하였다.

이들 각 모델의 구성을 간략히 살펴보면 다음과 같다.

## 2. 다층 퍼셉트론 모델

퍼셉트론 (perceptron) 은 그림 3-1에 표시한 내용의 구성을 갖는 신경회로망 소자로 정의된다.



$$\text{net} = \text{input1} + \dots + \text{inputN} + \text{bias}$$

$$\text{OUT} = \text{sigmoid}(\text{net})$$

그림 3-1. 퍼셉트론 소자의 모형도



이는 신경회로망 분야에서 가장 많이 응용되고 있는 것으로, 다층 퍼셉트론의 적용에서 충분한 입출력 상관관계 데이터가 있을 때, 가장 문제가 되는 것은 학습시간을 최소화하는 문제와 학습과정에서 오차곡면 ( error surface ) 의 국부적인 최소 ( local minimum ) 로 수렴하는 것을 방지하는 것이며, 국부적인 최소문제를 해결하기 위해서는 볼츠만 머신등의 통계적 방법을 이용하는 방법들이 연구되어 왔다.

학습시간을 줄이기 위한 방법들은 학습의 반복과정에서 오차곡면에서의 탐색방향과 그 벡터 크기를 결정하는 과정에서 차이가 나게 되는데, 대표적인 것으로는

(1) 학습의 반복과정에서 가중치(weight)의 변화량에 모멘텀 항을 포함시키는 방법으로서 이 모멘텀 항은 오차곡면 상에서의 low pass filter 역할을 하게 되며, 가중치 오차를 적분하여 가중치 변화를 결정하는 역할도 하는 것으로 알려져 있다.

(2) 델타 규칙 ( delta rule ) 과 이를 숨은층과 입력층에 적용하기 위한 역전파 ( back propagation ) 는 최적화 이론으로 보면 일차 미분을 사용한 steepest descent 방법이 이에 해당된다. 그러나 이 방법은 오차곡면  $Error(X) = c, c$  는 임의의 상수로 하였을 때 생기는 폐곡선이 eccentric 한 경우에는 수렴속도가 굉장히 느린것으로 알려져 있다. 학습과정의 오차곡면이 매우 복잡하므로 이러한 문제를 해결하기 위해 이차미분을 사용하는 방법들이 제시되고 있다.

그리고 그 이외에도

(3) 학습과정에 들어가기 전에 가중치의 초기치를 적절하게 조정하는 방법

(4) 학습과정 중에 네트워크 구조를 변경시키는 방법

등이 있다.

본 연구에서는 학습 데이터의 성격에 따라 일반적인 델타 규칙만을 적용하는 것과 위의 4가지 방법을 각각 적용하여 학습과정을 할 수행시켜 보았다. 그리고 다층 퍼셉트론 신경회로망은 Fig.2-2에 보이는 바와 같이 입력층, 두개의 숨은층, 그리고 출력층으로 구성하였다.

입력층으로는 가공 feature type, 절삭량 ( 절삭깊이, 절삭폭 ), 피삭재질 type, 공구재질 type로 이루어져 있으며, 출력층은 표준 이송속도 및 절삭속도이다. 이때 네트워크 학습과정을 효율적으로 하기 위해 입력층과 출력층에서는 각각 자체 스케일링이 되도록 하였다.

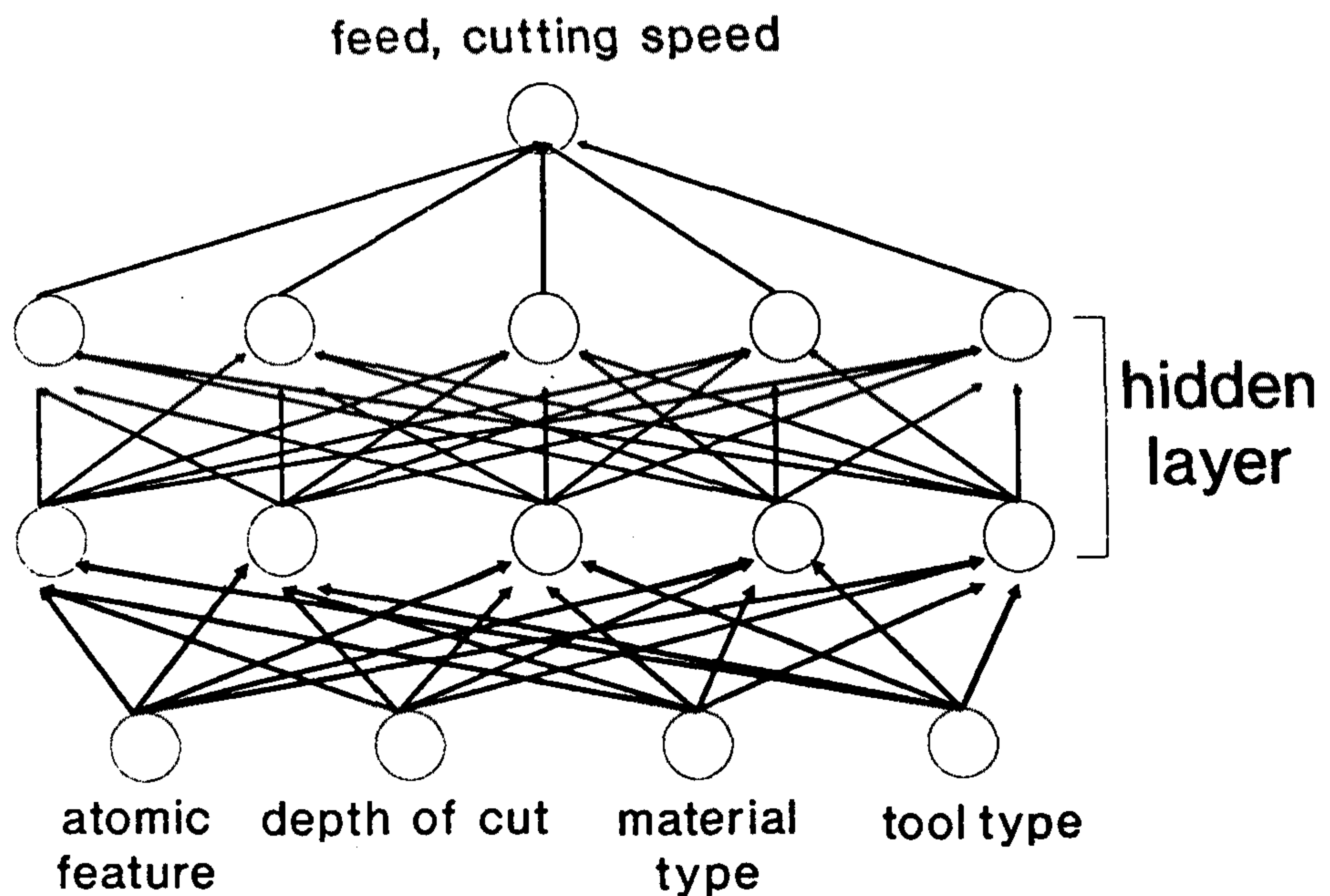


그림 3-2. 다층 퍼셉트론 신경회로망의 네트워크 구조



### 3. CMAC 신경회로망

한편 본 연구에서는 데이터의 불규칙성을 학습절차에 의해서 자체적으로 데이터 갱신이 되도록 하기 위해서 Albus에 의하여 제안된 CMAC(Cerebellar Model Articulation Controller) 신경회로망 모델이 부가적으로 사용되었는데, 이는 메모리 테이블을 사용하여 데이터가 갖는 비선형 관계를 부가적으로 나타내기 위한 것이었다.

신경물리학분야의 연구에 의하면 인간의 동작을 제어하는 것은 소뇌(cerebellum)로 알려져 있는데 소뇌에 입력되는 것은 어떠한 동작을 하라는 명령의 내용과 근육, 관절, 피부 등에서 감지되어 피드백 되어오는 입력이 있다. 그런데 연구 결과에 따르면 예를 들어 근육을 움직일 때, 이렇게 피드백 되어오는 입력정보는 그 감지된 크기를 갖는 것이 아니라 근육을 움직이는 신호를 갖고 있는 세포들의 위치 즉 메모리 주소로 구성되어 있는 것으로 알려졌다.

이러한 원리를 응용하여 신경 회로망을 구성한 것이 바로 CMAC인데, 입력 상태 벡터  $s$ 와 출력상태  $F(s)$ 간의 관계를 나타내는 CMAC의 기본 구조는 그림 3-3과 같다.

$N$ 차원의 입력공간  $S$ 의 지점인 입력상태 벡터  $s$ 는  $N$ 차원의 메모리  $A$ 의  $C$  갯수 만큼의 위치로 매핑된다. 출력상태 벡터  $F(s)$ 는 이 매핑된 위치의 메모리의 내용을 모두 더하여서 얻게 된다.  $A$ 의 메모리 구성에는 그 갯수가 굉장히 많이 소요되지만 이들 전체에는 한꺼번에 데이터가 입력되지 않고 뜨문 뜨문 데이터가 입력되게 된다. 따라서 메모리 갯수가 클 때에는 hash code 방법을 사용하여 보다 작은 크기의 메모리에 랜덤하게 매핑시키는 방법을 사용하게 된다.

두 입력 벡터의 해밍거리가  $C$ 보다 작게 되면 두 입력 벡터는 메모리를 공유하게 된다. 이 성질에 의해서 출력상태 벡터  $F(s)$ 는 입력 공간에서 비슷

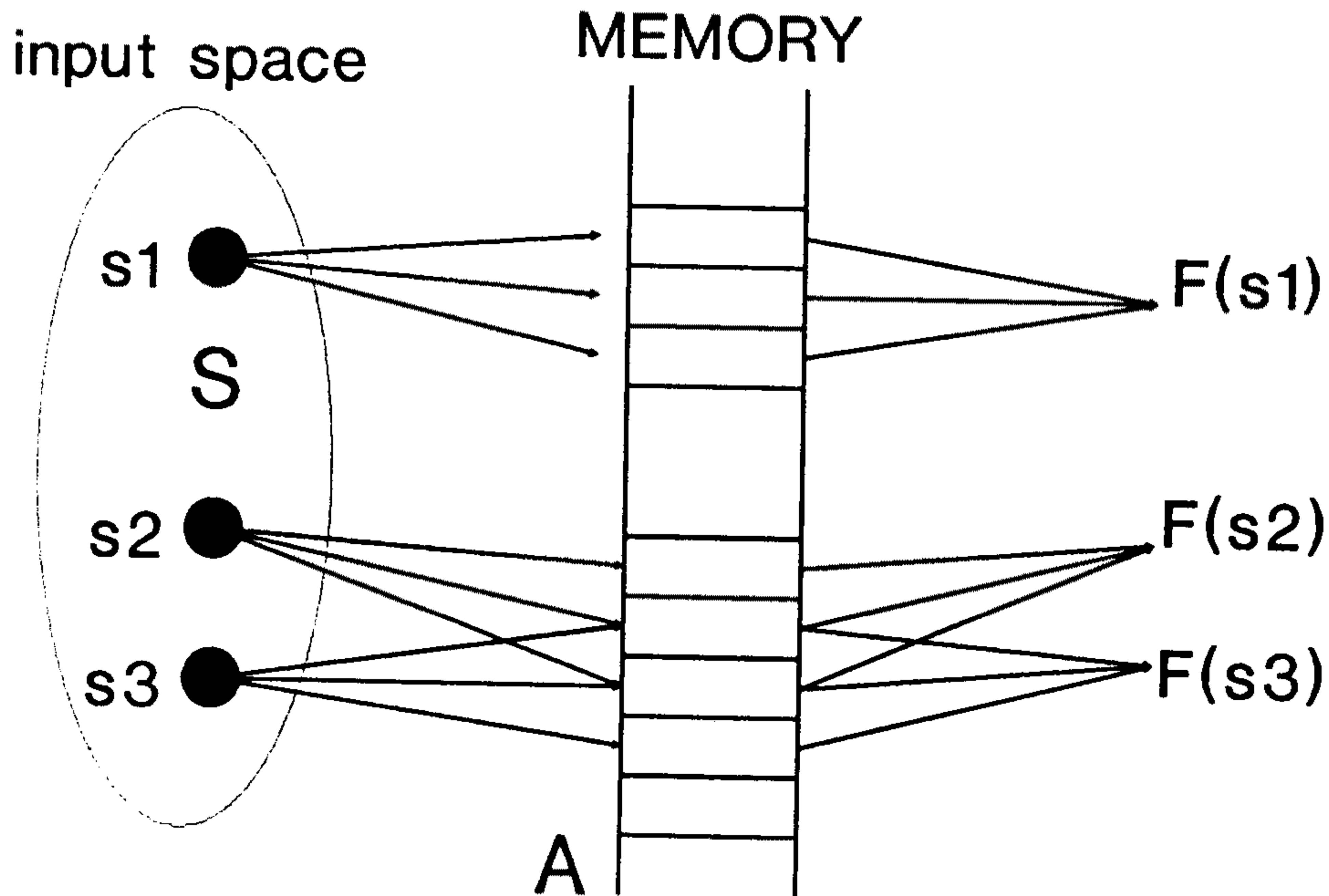


그림 3-3. CMAC 신경회로망의 계산모형

한 두 입력에 대하여 비슷한 출력을 내게 되는 일반화 ( generalization ) 의 특성을 갖는다.

입력 상태 벡터  $s$ 와 출력상태 벡터  $F(s)$ 의 실제 데이터가 주어졌을 때 메모리의 내용을 갱신하는 것은 다음의 규칙에 따른다.

$$\text{CONTENT}( A(s) )_k = \text{CONTENT}( A(s) )_{k-1} + \mu$$

$$\mu = o \cdot (F_0 - F(s))/C$$

여기서  $A(s)$  : 입력 상태 벡터  $s$  에서 매핑된 메모리

$\text{CONTENT}( A(s) )_k$  : 시간  $k$  스텝에서의  $A(s)$  메모리의 내용값

$\mu$  : 교정인자 (correction factor)

$o$  : 훈련인자 (training factor)  $0 < o < 1$

$F_0$  : 출력상태 벡터의 실제 값

$F(s)$  : 메모리에서 구해진 출력상태 벡터

$C$  : 이웃 영역 인자 (neighborhood factor)

#### 4. 퍼지 숫자에 의한 불확실성 처리 규칙

본 연구에서는 전문가로부터 수집된 절삭가공데이터의 성격이가 극히 가식적이거나, 그 불규칙 데이터의 오차가 극히 큰 경우의 대비책으로써 수집된 데이터의 신뢰확률이 Fuzzy 숫자의 형태로 반영될 수 있도록 하였고, 이들 Fuzzy값의 처리에 의해서 데이터가 안정되도록 하였으며, 또한 개선된 데이터들은 학습기능(learning performace)에 의해서 모형화 될 수 있도록 하였다.

여기서 사용된 처리 규칙을 간략히 살펴보면 다음과 같다.

일반적으로 퍼지집합은 다음과 같은 특성함수

$$\mu_A : X \rightarrow [0,1]$$

$$\text{또는 } \mu_A : X \rightarrow L$$

에 의하여 정의하게 되는데, 여기서  $L$  은 래티스 이다.

퍼지집합은 유한형태인 경우에는

$$A = \mu_A(x_1)/x_1 + \dots + \mu_A(x_n)/x_n = \sum_{i=1}^n \mu_a(x_i)/x_i$$

그 이외에는

$$A = \int \mu_A(x)/x.$$

으로 일반적으로 나타낸다. 이때 퍼지집합  $A$ 의 지원영역 (support)  $\text{supp } A$  는  $\text{supp } A = \{ x \in X, \mu_A(x) > 0 \}$  으로 나타내며, 퍼지집합  $A$ 의 높이  $\text{hgt}(A)$ 는  $\text{hgt}(A) = \{ \forall x \in A, \text{SUP } \mu_A(x) \}$  으로 정의되며, 퍼지집합  $\mu$

$\mu_A(X)$ 의 높이가 1 이면 정규화 ( normalized ) 되어 있다고 한다.

수준집합 ( level set )  $\Lambda_A$  는  $\Lambda_A = \{ \alpha \mid \mu_A(X) = \alpha \text{ for some } x \in X \}$  로 나타낸다.

퍼지 집합이 볼록 ( convex ) 하기 위한 조건은 다음과 같다.

$$\forall r \in X, \forall s \in X, \forall \lambda \in \{0, 1\}$$

$$\mu_A( \lambda r + (1-\lambda)s ) \geq \min( \mu_A(r), \mu_A(s) ).$$

또는 A의  $\alpha$ 절단이  $[a_1(\alpha), a_3(\alpha)]$  로 주어질 때

$$( \alpha' < \alpha ) \Rightarrow ( a_1(\alpha') \leq a_1(\alpha), a_3(\alpha) \leq a_3(\alpha') ) \Rightarrow ( A\alpha \subset A\alpha' )$$

퍼지집합  $\mu_A(X)$ 가 전체집합  $X = R$  에 대하여 정의되고, 정규화 되어 있으며 볼록하면 이를 퍼지숫자라 한다.

퍼지량에 대하여 기존의 퍼지하지 않는 수학 개념의 적용을 위해서는 퍼지집합의 확장원리가 적용되어야 하는데, 퍼지 숫자를 통한 연산은 이러한 확장원리를 이용하여 정의 하게 된다.

이상의 원리 적용을 간결하게 할 수 있는 퍼지숫자의 형태로써 본 연구에서는 삼각형 및 L-R 퍼지숫자가 사용되었다.

### (가) 삼각 퍼지숫자

삼각 퍼지숫자는  $A=(a_1, a_2, a_3)$ 로 표현하며, 소속함수는

$$\mu_A(X) = \begin{cases} 0 & \text{if } x < a_1 \\ (x-a_1)/(a_2-a_1) & \text{if } a_1 \leq x \leq a_2 \\ (a_3-x)/(a_2-a_1) & \text{if } a_2 \leq x \leq a_3 \\ 0 & \text{if } x > a_3 \end{cases}$$

으로 나타낸다. 그리고 연산은 다음과 같이 정의한다.

$$Aa = [ (a_2-a_1)a + a_1, -(a_3-a_2)a + a_3 ]$$

$$A + B = ( a_1+b_1, a_2+b_2, a_3+b_3 )$$

$$A - B = ( a_1-b_1 , a_2-b_2 , a_3-b_3 )$$

$$- A = ( -a_3 , -a_2 , -a_1 )$$

$$A * B = ( \text{left}(A_0*B_0), (A_0*B_0), \text{right}(A_0*B_0) )$$

$$A / B = ( \text{left}(A_0/B_0), (A_0/B_0), \text{right}(A_0/B_0) )$$

(나) L-R 퍼지숫자

L-R 퍼지숫자  $A=( m, \alpha, \beta )$  로 표현하며, 소속함수는

$$\mu A(x) = \begin{cases} L((m-x/\alpha)) & \text{if } x \leq m, \alpha > 0 \\ R((x-m/\beta)) & \text{if } x \geq m, \beta > 0 \end{cases}$$

나타난다. 여기서  $m$  은 평균,  $\alpha, \beta$ 는 폭, L,R은 참조함수인데,  $L(x)=L(-x)$ ,  $L(0)=1$ , 그리고  $[0, \infty]$  구간에서 단조 감소하는 성질을 갖고 있다.

연산은 다음과 같이 정의한다.

$$A + B = (m, \alpha, \beta)(+)(n, \gamma, \delta) = (m+n, \alpha+\gamma, \beta+\delta)$$

$$A - B = (m-n, \alpha+\delta, \beta+\gamma)$$

$$- A = (-m, \beta, \alpha)$$

$$A * B = (mn, m\gamma+n\alpha, m\delta+n\beta)$$

# 제 4 장 DB editor 의 작성과 데이터베이스의 구축

## 제 1 절 절삭성 데이터의 수집방법

### 1. Feature 별 class 의 구성방법

제 1차년도에서 개발된 메인 모듈에서 절삭가공 파라메타를 설정하는 과정은 앞의 그림 2-1과 같이 공정별 공구숫수, 절삭깊이, 절삭폭, 절삭속도, 이송속도, 절삭소요시간 등이 공유화일로 수행과정 중에 생성되고 있고, INTLFL 화일유지 프로그램에 의해서 관리되는 표준데이터로는 CTF, 공작기계, 재질, 공구, 샘플 절삭조건 등의 화일이 준비되어 있다.

이같은 데이터의 지배 환경하에서 절삭성데이터를 확보하기 위해서는 이들 파라메타들과 서로 유기적인 관계로 구성될 수 있는 데이터의 생성 프로그램이 필요하고, 앞의 3-1절에서 살펴 보았던 2단계의 매핑관계도 이들 환경하에서 공유되는 파라메타들으로써 데이터가 생성될 것이 요구된다.

그러나 feature별로 프로그램 수행과정 도중의 ( 기존의 논리으로써 간헐적으로 생성되는 ) 유기적 관계를 그대로 유지하면서 앞의 매핑관계를 적용하는 것은 1차년에 개발된 대부분의 source program을 수정해야 되는 매우 복잡한 일이 될 것이고, 그렇다고 해서 표준 절삭조건화일을 미세한 다변수의 구조로 구성시켜 현실감을 증진시킨다고 하더라도 이는 매우 방대한 데이터 화일을 두게 되어서 규모 면이나 실시간 이내에서의 수행면에서 비효율적이며 데이터의 신뢰도를 증진시키는 데에도 한계가 있게 된다.

그 때문에 2차년 연구에서는 앞의 매핑관계 중에서 제 1단계로 절삭폭과



절삭깊이 등의 기하학적 가공인자를 결정하는 것은 COPS에서 각 feature 별로 사용하는 과정을 그대로 살려서 수행하는 것으로 하였고, 한편 제 2단계에서 동적 가공제어인자를 결정하는 과정은 COPS에서 2개의 subroutine ( FENDS 및 CUT ) 만을 수정하는 것으로 가능하므로 절삭계수로 구성된 데이터베이스 ( 전문가로부터 수집된 절삭성데이터를 DB editor에 입력하여 처리된 출력 데이터베이스 ) 를 만들어 설치하고, 이를 통해 동적 가공제어인자를 결정하는 과정은 수정된 2개의 subroutine이 대처해 나가는 것으로 하였다.

한편 앞의 3장에서의 매핑관계 구성 중에 Class를 나누는 데에는 공작기계와 절삭가공 feature ( 가공 feature 및 공구 category ) 들도 포함되어 있는데, 이들 파라메타들이 포함되어 있을 때에는 다른 가공제어인자들을 정의하는 방법이 경우에 따라서 차이가 생기고 그 규격을 유한 갯수로 표현하기가 또한 어려울 뿐만 아니라 경우 수가 너무 많아서 데이터베이스가 너무 대규모화 되는 예로점이 있다. 그 때문에 공작기계의 종류와 가공 한계 범위는 modify rule를 통하여 대처해 나가고, 가공 feature에 있어서는 각각의 경우에 대해서 독립적으로 class를 구성해 나가는 것으로 모색하였다.

따라서 가공 feature를 앞의 그림 2-2와 같이 Face Milling, Perpheral End Milling, Slotting & Pocket Milling, Drilling, Boring, Reaming, Counterboring & Spotfacing, Tapping 의 8가지로 구성시켰고 가공형상특징과 공구 category에 일대일 대응이 될 수 있도록 매칭시켰다.

이들 각각의 가공 feature들에 대한 class는 절삭성의 공리화 과정을 토대로 모델링하였는데, 먼저 금형용 재질에 대한 class의 구분은 표 4-1과 같이 총 30가지로 정하였다.



표 4-1. 금형용 재질의 30가지 class 구분 내용과 규격.

가공물 재질 CLASS X1	한국공업규격		경도 ( BHN )	외국관련규격		
	기호	규격 No. 및 명칭		JIS	AISI / SAE	DIN
1	STD8 STD61	KSD3753 합금공구강 강재	Rc 50 - 55	SKD8 SKD61	H19 H13	X40CrMoV51
2	STD6 STD62	KSD3753 합금공구강 강재	Rc 48 - 50	SKD6 SKD62	H11 H12	X38CrMoV51
3	STD1 STD11	KSD3753 합금공구강 강재	350 - 400	SKD1 SKD11	D3 D2	X210Cr12
4	SM09CK SM15CK SM20CK SUP3 SUP5 SUP11	KSD3752 기계구조용 탄소강 강재, 스프링강	375 - 425	S09CK S15CK S20CK SUP3 SUP5 SUP11	1011 1015 1020 1541 1552 1547	CK10 CK15 CK22
	SNM616 SNM625 SNM630 SNM815	KSD3709 니켈크롬 물리브덴강 강재		SNM616 SNM625 SNM630 SNM815		
	SNM220H SNM420H SNC631H SNC815H	KSD3754 경화능보증 구조용 강재 <H강>		SNM220H SNM420H SNC631H SNC815H	8617H, 8620H 8622H, 4320H	
5	SKH5 SKH10 SKH53 SKH57 SKH58 SKH59	KSD3522 고속도공구강 강재	350 - 400	SKH5 SKH10 SKH53 SKH57 SKH58 SKH59	T6 T15 M3-2 M7 M42	S6-5-3 S10-4-3-10 S2-10-1-8
6	SNM431 SNM439 SNM447	KSD3709 니켈크롬물리 브덴강 강재	325 - 375	SNM431 SNM439 SNM447	4340	
	SCM440 SCM445 SCM822	KSD3709 크롬 물리 브덴강 강재		SCM440 SCM445 SCM822	4140, 4142 4145, 4147	42CrMo4

가공물 재질 CLASS XI	한국공업규격		경도 ( BHN )	외국관련규격		
	기호	규격 No. 및 명칭		JIS	AISI / SAE	DIN
7	SKH3 SKH4 SKH54 SKH55 SKH56	KSD3522 고속도공구강 강재	325 - 375	SKH3 SKH4 SKH54 SKH55 SKH56	T4 T5 M4 M36	S6-5-2-5
8	SCM415H SCM418H SCM420H SCM435H SCM440H SCM445H SCM822H SNC415H	KSD3754 경화능보중 구조용 강재 <H강>	325 - 375	SCM415H SCM418H SCM420H SCM435H SCM440H SCM445H SCM822H SNC415H	4135H, 4137H 4140H, 4142H 4145H, 4147H	34CrMo4 42CrMo4
9	SKH2 SKH9 SKH51 SKH52	KSD3522 고속도공구강 강재	275 - 325	SKH2 SKH9 SKH51 SKH52	T1 M2 M3-1	S6-5-2
10	STD4 STD5 STS3 STS31 STS93 STS94 STS95	KSD3753 합금공구강 강재	250 - 325	SKD4 SKD5 SKS3 SKS31 SKS93 SKS94 SKS95		105WCr6
11	SNM415 SNM420 SNM431	KSD3709 니켈크롬몰리 브덴강 강재	220 - 300	SNM415 SNM420 SNM431	4320	
	SCM415 SCM418 SCM420 SCM421	KSD3711 크롬몰리브덴 강 강재		SCM415 SCM418 SCM420 SCM421		
	SMn420 SMn433 SMn438	KSD3724 기계구조용 망간강 강재		SMn420 SMn433 SMn438	1522 1536 1541	
	SCr440 SCr445	KSD3707 크롬강 강재		SCr440 SCr445	5140 5145	41Cr4

가공물 재질 CLASS X1	한국공업규격		경도 ( BHN )	외국관련규격		
	기호	규격 No. 및 명칭		JIS	AISI / SAE	DIN
12	STS347 STS384 STSM7 STSM15J1	KSD3697-3706 스테인레스강 오오스테나이트계	180 - 250	SUS347 SUS384 SUSXM7 SUSXM15J1	AISI1347 AISI1384 ASTMXM7 ASTMXM15	DINX6CrNi- Nb1810
	SCS14 SCS18 SCS23 SCS24	KSD3697-3706 스테인레스강 주강품		SCS14 SCS18 SCS23 SCS24		
13	STS5 STS7 STS8 STS51	KSD3753 합금공구강 강재	200 - 250	SKS5 SKS7 SKS8 SKS51	L6	
14	STS2 STS11 STS21	KSD3753 합금공구강 강재	200 - 250	SKS2 SKS11 SKS21	F2	105WCr6
	STS309S STS310S STS316 STS317	KSD3697-3706 스테인레스강 오오스테나이트계		SUS309S SUS310S SUS316 SUS317	AISI309S AISI310S AISI316 AISI317	
15	STC5 STC6 STC7	KSD3751 탄소공구강 강재	225 - 260	SK5 SK6 SK7	W1-8 W1-7	C80W1 C60W3
16	SNC236 SNC415 SNC631 SNC815 SNC836	KSD3708 니켈크롬강 강재	150 - 250	SNC236 SNC415 SNC631 SNC815 SNC836		
	STS201 STS202 STS301 STS302 STS303 STS304 STS305 STS321	KSD3697-3706 스테인레스강 오스테나이트계		SUS201 SUS202 SUS301 SUS302 SUS303 SUS304 SUS305 SUS321	AISI201 AISI202 AISI301 AISI302 AISI303 AISI304 AISI305 AISI321	6CrNiTi1810

가공물 재질 CLASS X1	한국 공업 규격		경도 ( BHN )	외국 관련 규격		
	기호	규격 No. 및 명칭		JIS	AISI / SAE	DIN
17	SM50C SM53C SM55C SM58C	KSD3752 기계구조용 탄소강 강재	150 - 250	S50C S53C S55C S58C	1049 1050, 1053 1055 1060	
	STS4 STS41 STS43 STS44	KSD3753 합금공구강 강재		SKS4 SKS41 SKS43 SKS44	W2-9 W2-8	
18	STS403 STS410 STS416 STS420	KSD3697-3706 스테인레스강 마르텐사이트계	175 - 225	SUS403 SUS410 SUS416 SUS420	AISI403 AISI410 AISI416 AISI420	10Cr13  20Cr13
19	STS429 STS431 STS440A STS440B STS440C STS440F	KSD3697-3706 스테인레스강 마르텐사이트계	175 - 225	SUS429 SUS431 SUS440A SUS440B SUS440C SUS440F	AISI431 AISI440A AISI440B AISI440C ASTM440F	20CrNi172
20	SM40C SM43C SM45C SM48C	KSD3752 기계구조용 탄소강 강재	190 - 225	S40C S43C S45C S48C	1039, 1040 1042, 1043 1045, 1046	CK45, C45
21	SMn443 SMnCr420 SMnCr443 SAICrMo1	KSD3724, 3756 기계구조용 망간크롬강	175 - 225	SMn443 SMnCr420 SMnCr443 SACM645	1522	41CrAlMo7
22	SM30C SM33C SM35C SM38C	KSD3752 기계구조용 탄소강 강재	175 - 225	S30C S33C S35C S38C	1030  1035, 37 1038	CK35, C35
23	SM20C SM22C SM25C SM28C	KSD3752 기계구조용 탄소강 강재	160 - 200	S20C S22C S25C S28C	1020 1023 1025 1029	CK20, C20 CK22, C22
24	SM10C SM12C SM15C SM17C	KSD3752 기계구조용 탄소강 강재	120 - 200	S10C S12C S15C S17C	1010 1012 1015 1017	CK10, C10 CK15, C15

가공물 재질 CLASS X1	한국공업규격		경도 ( BHN )	외국관련규격		
	기호	규격 No. 및 명칭		JIS	AISI / SAE	DIN
25	STS405 STS410L STS429 STS430 STS434 STS436L STS444	KSD3697-3706 스테인레스강 페라이트계	135 - 185	SUS405 SUS410L SUS429 SUS430 SUS434 SUS436L SUS444	AISI405  AISI429 AISI430 AISI434 AISI436L AISI444	6CrA113  6Cr17 6CrNb17 6CrMo171
26	STC1 STC2 STC3 STC4	KSD3751 탄소공구강 강재	100 - 200	SK1 SK2 SK3 SK4	W1-13 W1-11 W1-10 W1-9	C105W1
27	YBsC BsW BC PBC	황동각재 황동선재 청동각재 인칭동	100 -150	YBsC BsW BC PBC		
28	BSBME BSBMD KJ KPb	궤삭황동  구리납합금 (베어링용)	40 -175	BSBME BSBMD KJ KPb		
29	ADC6 ADC7 ADC8 ADC10 ADC12	알루미늄합금 다이캐스팅재	50 -125	ADC6 ADC7 ADC8 ADC10 ADC12		
30	ADC1 ADC2 ADC3 ADC4 ADC5	알루미늄합금 다이캐스팅재	50 -125	ADC1 ADC2 ADC3 ADC4 ADC5		



이 재질 클래스는 X1의 가공인자변수로서 모든 가공 feature들에 대하여 공통적으로 사용되도록 하였으며, X1의 크기는 5인의 전문가가 응답한 절삭속도값을 기준으로 이들 값이 비슷하게 응답되는 재질들을 먼저 같은 그룹으로 묶어서 정리하였고, 그 다음에 각각의 그룹들을 절삭속도값의 크기 순서대로 클래스 값을 부여하였다.

여기에는 현재 국내에서 사용되고 있는 사출 및 프레스 금형용의 재료들을 총 망라하여 적용하였고, 이들 재료는 또한 한국공업규격과 외국관련규격으로 동시에 표현하여 해당재질이 쉽게 검색되어 해당 클래스가 결정될 수 있도록 하였다. 그리고 이들을 정리하는 과정에 있어서는 유연하게도 장치 modify rule에서 사용되는 비절삭저항, 비절삭동력 등의 절삭상수들도 비슷한 크기의 순서를 갖는 것으로 판명되었으며, 그 때문에 이 재질 클래스 구분은 매핑과정에 있어서의 수렴성과 탈 local minimum 특성에 효과적으로 사용될 수 있었다.

각 가공 feature 별로 ( X2, X3, X4 ) 들에 관한 가공제어인자 변수들의 명칭과 클래스 구분들도 다음과 같이 구분하여 사용되었다.

절삭공구 ( X2 ) 에 대한 클래스 구분은 같은 공구재료라고 하더라도 절삭공구의 각 제조 메이커마다 조금씩 그 절삭성이 다르고 또한 공구 형상 및 홀더, 공구형상들에 따라서 절삭성의 차이가 있기 때문에 이를 미세하게 잡는 것은 데이터의 규모와 수렴성의 측면에서 무의미하였다. 그 때문에 여기서도 공구의 구체적인 특성 차이에 대한 사항은 장치의 modify rule에 의해서 대처해 나가기로 하고 여기서는 그 차이를 다소 명확하게 구분할 수 있는 4가지의 클래스로 나누었다. 클래스1은 SK, SKS, SKD, SKH, SKT 등의 계열에 속하는 고속도강재의 공구를 포함시키는 것으로 하였으며, 클래스2는 P, M, K계에 해당하는 코팅되지 않은 brazed type의 초경재 공구들이 포함되는 것으로 하였고, 클래스3은 인서트 타입 중심의 코팅된 초경재 공구를 포함시켰으며, 클래스4는 서멧트재, 세라믹재, CBN, Diamond 툴 계열들

이 포함되는 것으로 하였다.

그러나 이들 공구재질 및 형상 클래스는 face milling, peripheral end milling, slotting & pocketing 의 3가지 가공 feature에만 적용되는 것으로 하였고, drilling에 있어서는 이와는 달리 공구의 형상에 대한 특성 차도 고려하여 취급하였다. 그래서 twist HSS 계열의 드릴공구는 클래스1에 모두 포함시켰으나, oil hole 드릴에 있어서는 고속도강재와 초경재를 함께 많이 사용하고 있고 두 재질의 가공특성도 많이 차이를 나타내기 때문에 각각 클래스2 및 클래스3로 나누어 구분하였고, gun 드릴에 있어서는 앞의 일반적인 드릴들과 그 특성이 다르므로 클래스4에 포함시켰다.

또한 boring에 있어서는 보링바에 인서트되는 팁을 기준으로 HSS재 bore를 클래스1에, WC brazed bore와 uncoated WC bore는 클래스2에, indexable uncoated WC bore는 클래스3에 두었으며, CP계 초경 보어 및 세라믹재 보어는 클래스4에 포함시켰다.

Reaming에 있어서는 다듬질가공의 경우와 황삭의 경우에 그 특성 차이가 크므로 고속도강재로 다듬질가공 하는 경우를 클래스1에, 초경재로 다듬질 하는 경우는 클래스2에, 그리고 각각의 재질에 대해 황삭가공하는 경우는 클래스3 및 클래스4로 구성시켰다.

또 카운터보링 및 스폿페이싱의 경우에는 대개 가공이 단순화되어 있어서 고속도강재를 사용하는 경우와 초경재를 사용하는 경우의 2가지로 구분하였고, tapping의 경우에는 공구에 대한 클래스 구분이 불명확하여 취급하지 않았다.



[1] Face Milling ( X1, X2, X3, X4 ) ----> ( Y1, Y2 )

가공인자	X2		X3		X4	
명 칭	공구재질 및 형상		절삭 깊이		절삭 폭	
클 래 스	1	고속도강 재 ( HSS: SK, SKS, SKD, SKH, SKT 계 )	1	1 mm 기준	1	D / 8 (mm)
	2	Uncoated 초경재 ( P, M, K계, Brazed Braze type 중심 )	2	2 mm	2	D / 4 (mm)
	3	Coated 초경재 ( P, M, K계, Insert type 중심 )	3	4 mm	3	D / 2 (mm)
	4	Cermet, Ceramic, CBN, Diamond 를 계열	4	8 mm	4	2D / 3 (mm)
Y 1	절삭 속도 ( m / min. )					
Y 2	이송량 ( mm / tooth )					

[2] End Milling - Peripheral ( X1, X2, X3, X4 ) ----> ( Y1, Y2 )

가공인자	X2		X3		X4	
명 칭	공구재질 및 형상		반경방향 절삭 깊이		커터 직경	
클 래 스	1	고속도강 재 ( HSS: SK, SKS, SKD, SKH, SKT 계 )	1	0.5 mm 기준	1	10 mm
	2	Uncoated 초경재 ( P, M, K계, Brazed Braze type 중심 )	2	1.5 mm	2	12 mm
	3	Coated 초경재 ( P, M, K계, Insert type 중심 )	3	D / 4 (mm)	3	18 mm
	4	Cermet, Ceramic, CBN, Diamond 를 계열	4	D / 2 (mm)	4	50 mm
Y 1	절삭 속도 ( m / min. )					
Y 2	이송량 ( mm / tooth )					

[3] End Milling - Slotting & Pocketing ( X1, X2, X3, X4 ) --> ( Y1, Y2 )

가공인자	X2		X3		X4	
명 칭	공구재질 및 형상		축방향 절삭깊이		슬롯의 폭	
클 래 스	1	고속도강 재 ( HSS: SK, SKS, SKD, SKH, SKT 계 )	1	0.75 mm 기준	1	10 mm
	2	Uncoated 초경재 ( P, M, K계, Brazed Brazed type중심)	2	3 mm	2	12 mm
	3	Coated 초경재 ( P, M, K계, Insert type중심)	3	D / 2 (mm)	3	18 mm
	4	Cermet, Ceramic, CBN, Diamond 등 계열	4	D / 1 (mm)	4	50 mm
Y 1	절삭 속도 ( m / min. )					
Y 2	이송량 ( mm / tooth )					

[4] Drilling ( X1, X2, X3 ) ----> ( Y1, Y2 )

가공인자	X2		X3		Y 1	
명 칭	공구재질 및 형상		구멍 직경		절삭 속도 ( m / min. )	
클 래 스	1	Twist HSS Drill ( HSS: SK, SKS, SKD, SKH, SKT 계 )	1	1.5 mm	Y 2	
			2	3.0 mm		
	2	Oil Hole HSS Drill , WC Spade Drill	3	6.0 mm		이송량 ( mm / rev. )
			4	12 mm		
	3	Oil Hole WC Drill, WC(K계) Drill	5	18 mm	<비고> Gun drill의 경우 $X3 = X3 / 3.5$ 로 상사됨.	
			6	25 mm		
	4	Gun Drill	7	35 mm		
			8	50 mm		

[5] Boring ( X1, X2, X3 ) ----> ( Y1, Y2 )

가공인자	X2		X3		Y 1
명 칭	공구재질 및 형상		절삭 깊이		절삭 속도 ( m / min. )
클 래 스	1	HSS 재 Bore ( HSS: SK, SKS, SKD, SKH, SKT 계 )	1	0.2 mm	Y 2
	2	WC Brazed Bore, Uncoated WC Bore	2	0.5 mm	이송량 ( mm / rev. )
	3	Indexable uncoated WC(K계) Bore	3	1.25 mm	
	4	CP계 Coated WC 및 Ceramic 재 Bore	4	2.5 mm	

[6] Reaming ( X1, X2, X3 ) ----> ( Y1, Y2 )

가공인자	X2		X3		Y 1
명 칭	공구재질 및 형상		리이머 직경		절삭 속도 ( m / min. )
클 래 스	1	HSS 재 Reamer - Finishing ( 다듬질 )	1	3 mm	Y 2
			2	6 mm	
	2	WC 재 Reamer - Finishing ( 다듬질 )	3	12 mm	이송량 ( mm / rev. )
			4	25 mm	
	3	HSS 재 Reamer - Roughing ( 황삭 )	5	35 mm	
			6	50 mm	
	4	WC 재 Reamer - Roughing ( 황삭 )			

[7] Counter Boring & Spotfacing ( X1, X2, X3 ) ----> ( Y1, Y2 )

가공인자	X2		X3		Y 1	
명 칭	공구재질 및 형상		공칭 구멍 지름		절삭 속도 ( m / min. )	
클 래 스	1	HSS 재 Counter-Bore 및 Spotfacer	1	6 mm	Y 2	
			2	12 mm		
	2	WC 재 Counter-Bore 및 Spotfacer	3	18 mm	이송량 ( mm / rev. )	
			4	25 mm		
				5	35 mm	
				6	50 mm	
				7	62 mm	
				8	75 mm	

[8] Tapping ( X1, X2 ) ----> ( Y1 )

가공인자	X2		Y 1
명 칭	나사 피치		절삭 속도 ( m / min. )
클 래 스	1	3 mm	
	2	2 mm	
	3	1 mm	
	4	0.5 mm	

## 2. 설문작성 및 수집데이터

본 연구에서 매핑관계를 구성하는 DB editor에는 전문가로부터 수집된 많은 절삭성데이터들이 입력되어야 하며, 이들 데이터들은 또한 그 신뢰성이 어느 정도는 보장이 되어 있어야 나중에 DB editor를 수행시켜 절삭계수들에 관한 데이터베이스를 구축할 때에 그 수렴성과 변수들의 normalize 가 가능해진다.

이러한 성격의 수집데이터들을 확보하기 위해서 본 연구에서는 금형 가공 현장에서 절삭가공 작업에 대한 경험이 5년 이상되는 실무자들 150명을 대상으로 1차 설문 및 응답 절차를 거쳤다.

제 1차 설문조사에서는 표 4-2 및 표 4-3에서 보이는 바와 같이 국내의 금형업체들에서 현재 사용하고 있는 공작기계, 절삭공구, 금형용재료들에 대한 데이터의 수집과 가공작업을 공정설계 또는 실제로 수행하는 데에 있어서의 가공상태 인자들, 그리고 각 인자들의 비중도에 관한 데이터를 수집하기 위한 것이었다.

그리고 공작기계, 절삭공구, 가공물재질 등에 관한 응답지의 내용은 그림 4-1과 같이 장차 구축될 데이터파일의 내용을 토대로 항목들이 제시되어 있다.

가공작업의 설계 또는 실행상의 고려 사항에 있어서는 표 4-4에 제시된 바와 같이 공작기계, 피삭재의 취부, 절삭공구, 가공작업에의 관련성 등으로 구분하여 설문하였으며 표 3-4에는 응답된 결과들을 정리한 내용도 함께 제시되어 있다. 여기서 응답내용난의 수치는 응답된 결과를 10을 기준으로 한 빈도수를 나타낸 것이고, 기타사항의 기입난에 쓴 내용은 응답된 결과 중에 많은 작업자들이 비슷한 성격으로 기입되었던 특이사항들을 적어 놓은 것이다.

한편 제 2차 설문조사에서는 1차설문 조사에서 나타났던 결과들을 토대로 실용적인 표준 절삭조건을 수집하는 조사였는데, 이 조사에서의 설문양식에는 그림 3-2와 같이 각각의 feature별로 양식을 작성하여 조사하였다.

그런데 이때에는 그 응답 데이터의 양이 너무 많고, 또한 응답된 내용들의 신뢰성이 너무 빈약한 것으로 판단되어 조사방법을 바꿨다. 그래서 변경된 조사에서는 절삭작업 현장에서만 15년 이상 종사하여 매우 그 경험이 풍부한 사람으로 인정되는 5인의 전문가들만을 추천받아 이들에 대해서만 설문조사된 데이터들을 정리하여 수집데이터로 정리하게 되었으며, 또한 설문양식에 있어서는 이들이 데이터를 작성하기 쉽도록 핸드북, 카타록, 가이드북 등에서 제시되고 있는 데이터들도 참조 데이터로 양식에 병기하거나 별도의 자료로 제공하였다.

정리된 수집데이터의 규모는 1인당 A4 용지로 약 300페이지에 달했으며, 총 수집데이터는 약 1500페이지에 이르렀다.

이같이 수집된 데이터는 앞에서 각 가공 feature 별로 구분된 class 값들의 변수와 더불어 수집 데이터베이스로 구축되었는데, 이때에는 데이터 테이블 작성용으로 Informix-SQL, ESQL/C 관계형 데이터베이스 구축 툴(AT&T 사)이 이용되었다. 그리고 그림 4-3은 이때 작성된 수집데이터 베이스의 face milling의 경우에 대한 한 예를 보여 준 것이며 이들에 대한 나머지 feature들에 대한 내용은 [ 부록 I ]에 제시된 바와 같다.



표 4-2. 제 1차 설문지의 조사목적 및 적용 범위

조사목적	상자형 공작물(금형부품)을 가공하는 데에 필요한 절삭가공 데이터 베이스를 우리나라의 실정에 맞도록 현실감 있게 구축하는 것임.	
적용범위	가공물 형상	상자형(각주형) 모양을 갖는 금형부품 또는 기계부품
	해당 가공형태	면삭가공, 측면밀링, 포켓밀링, 슬롯밀링, 구멍가공, 나사 가공, 카운터보링, 카운터싱킹, 방전가공, 연삭가공, 그리고 기타 이와 유사한 계통에 해당하는 가공들.
	해당 절삭공구	Face cutter, 엔드밀, 총형 커터류, 드릴계통, 보링공구, 리이어, 탭, 카운터보어, 카운터싱커, 방전기 전극류, 연삭숫돌 및 기타 유사한 절삭공구류.
	해당 공작기계	밀링머신, 드릴링머신, 보링머신, 머시닝센터, 보링센터, 방전가공기, 연삭기 및 기타 유사한 공작기계류. ( 범용기, 전용기, CNC기계 모두 포함됨 )

표 4-3. 제 1차 설문지의 조사대상별 작성 요령

조사대상	작성요령
(1) 공작기계	귀사에서 위의 가공형태의 작업에 동원될 수 있는 보유 공작기계에 대해서 해당항목별로 예시와 같이 기입해 주십시오. <주의> 단위를 꼭 확인하여 주시고, 부족하면 뒷면을 사용하시거나 별지를 사용하여 기입해 주셔도 좋습니다.
(2) 절삭공구	마찬가지로 귀사에서 사용 가능한 절삭공구들을 예시와 같이 가급적 모두 기입해 주시면 고맙겠습니다. 너무 양이 많으면 참고될 수 있는 목록표의 복사물을 첨부하셔도 좋습니다.
(3) 가공물재질 (또는 금형용재질)	귀사에서 지금까지 구입하여 사용한 경험이 있는 가공물 재질을 총망라하여 기입해 주십시오. 너무 양이 많으면 목록표 사본도 좋으며, 잘 모르는 항목은 공백으로 두셔도 좋습니다.
(4) 가공작업의 설계 또는 실행상의 고려사항	귀하가 위의 적용범위에 해당하는 절삭작업을 준비하거나 실제 수행한다고 생각하여, 표에서 주어진 항목에 대해 고려하는 지의 여부를 응답해 주십시오. 응답은 해당 부분에 "0 표"를 하시면 됩니다. 그리고 표에서 주어진 항목 이외에 고려할 수 있는 사항이 있다고 생각하시는 것은 무엇이든 "기타 항목" 란에 서술형으로 기입해 주시기 바랍니다. 또한 이 부분에 있어서는 현장에서 매우 경험이 많으신 기술자분들과 상의 하셔서 작성하시면 저희들에게 매우 도움이 됩니다.



(1) 공작기계

공작기계명칭 및 구입년도	제작회사 및 모델명	최대 주축 마력수 (kW)	최고 가공 정밀도 (mm)	이송 행정 (가로X세로) (mm x mm)	금속이송 속도(동상) (mm/분)	스핀들 회전수범위 (rpm)	이송속도범위 (밀링:mm/분, 드릴링:mm/rev)
예시 수직형 밀링 미신 (1983년)	대우중공업 GM-DCCE	12	0.005	1000x500	3000	20 - 4000	6.0 - 750

(2) 절삭공구

공구 명칭	제작(구입)회사	모델명 (제작 코드)	공구 type (Insert/ Blazed 등)	재질 및 규격(KS/JIS.. 등)	거터직경 (mm)	생크부 직경 (mm)	절삭날 수 (개/rev)	Helix 각 (°)	기타 특징
예시 엔드밀	한국야금	ZST9102b	Blazed타입	초경 P20	φ16	φ15	4 날	30°	
Face밀	SANDVIK	sv23-34 (NC몰링)	Insert	초경GC315 (코팅)	φ28	BT30규격	12날 팁	20° 외주각	삼각형팁/ 더블 양각형
GUN 드릴	TOSHIBA	DR76012	슬리드	HSS/SKH5	φ12	φ11	2 날	0°	Notch type

(3) 가공물 재질 ( 또는 금형용 재질 )

재질 명칭	제작(구입)회사	사용 용도	경도범위 (Hrc)	JIS 규격	AISI 규격	구입시 형상	담금질처리	기타 특징
예시 탄소구조강	日立金属	구조부품	45 - 60	SCM45	SM45C	각재	뜨임후 풀림	고장력강 임
금형공구강	한국중공업	금형용	64 - 75	SKH12	9002-M5	봉재	뜨임/풀림	쇄삭강 임

(4) 가공작업의 설계 또는 실행상의 고려사항

순서	구분	고려 사항	응답 내용 (해당년에 0 표)		
			항상 고려한다.	가끔 고려한다.	전혀 고려치 않는다.
1	공작기계 관련된 사항	1 공작기계의 정밀도는 만족되는가?			
		2 공작기계 스핀들의 강성은 충분인가?			
		3 공작기계의 동력은 이 가공에 충분인가?			
		4 공작기계가 가공중 진동을 일으킬까?			
		기타의 사항이			

그림 4-1. 제 1차 설문의 응답지 양식과 세부 항목

공작물재질	경도 (BHN)	표준 절삭 조건				추천(기준) 표준 공구분류 기호		
		반경방향 절삭깊이 (mm)	절삭속도 (m/min)	이송량 (mm/tooth)				
				커터 반경 (mm)				
10	12	18	25-50					
기 계 STK 30, 41, 51 등	225	.5						
	이하	1.5						
		D/4						
		D/2						
구 조 용 S30C --> S35C	175 - 225	.5						
		1.5						
		D/4						

공작물재질	경도 (BHN)	표준 절삭 조건			추천(기준) 표준 공구분류 기호
		절삭깊이 (mm)	절삭속도 (m/min)	이송량 (mm/tooth)	
기 계 STK 30, 41, 51 등	225	1	( 180 )	( 0.2 )	P 10
	이하	4	( 150 )	( 0.3 )	P 20
		8	( 135 )	( 0.4 )	
구 조 용 S30C --> S35C	175 - 225	1	( 171 )	( 0.2 )	
		4	( 146 )	( 0.3 )	
		8	( 125 )	( 0.4 )	
탄 소 S40C --> S58C	225 - 275	1	( 152 )	( 0.2 )	
		4	( 126 )	( 0.3 )	

그림 4-2. 제 2차 설문의 가공 feature별 응답지 양식과 세부항목

[ Type (1) Data in Face Milling Operations for Learning Procedure ]

Columns : X1 X2 X3 X4 Y1 Y2

No. of Data List = 1920

1.000	1.000	1.000	1.000	1.000	56.000	0.100
2.000	1.000	1.000	1.000	2.000	53.000	0.080
3.000	1.000	1.000	1.000	3.000	50.000	0.060
4.000	1.000	1.000	1.000	4.000	47.000	0.040
5.000	1.000	1.000	2.000	1.000	52.000	0.130
6.000	1.000	1.000	2.000	2.000	48.000	0.100
7.000	1.000	1.000	2.000	3.000	45.000	0.080
8.000	1.000	1.000	2.000	4.000	41.000	0.060
9.000	1.000	1.000	3.000	1.000	47.000	0.150
10.000	1.000	1.000	3.000	2.000	43.000	0.120
11.000	1.000	1.000	3.000	3.000	40.000	0.100
12.000	1.000	1.000	3.000	4.000	36.000	0.080
13.000	1.000	1.000	4.000	1.000	42.000	0.180
14.000	1.000	1.000	4.000	2.000	38.000	0.140
15.000	1.000	1.000	4.000	3.000	34.000	0.120
16.000	1.000	1.000	4.000	4.000	30.000	0.100
17.000	1.000	2.000	1.000	1.000	49.000	0.120
18.000	1.000	2.000	1.000	2.000	45.000	0.100
19.000	1.000	2.000	1.000	3.000	47.000	0.080
20.000	1.000	2.000	1.000	4.000	38.000	0.060
21.000	1.000	2.000	2.000	1.000	43.000	0.150
22.000	1.000	2.000	2.000	2.000	40.000	0.120
23.000	1.000	2.000	2.000	3.000	37.000	0.100
24.000	1.000	2.000	2.000	4.000	34.000	0.080
25.000	1.000	2.000	3.000	1.000	38.000	0.170
26.000	1.000	2.000	3.000	2.000	34.000	0.140
27.000	1.000	2.000	3.000	3.000	31.000	0.110
28.000	1.000	2.000	3.000	4.000	28.000	0.080
29.000	1.000	2.000	4.000	1.000	31.000	0.190
30.000	1.000	2.000	4.000	2.000	28.000	0.160
31.000	1.000	2.000	4.000	3.000	25.000	0.140
32.000	1.000	2.000	4.000	4.000	22.000	0.120
33.000	1.000	3.000	1.000	1.000	41.000	0.150
34.000	1.000	3.000	1.000	2.000	37.000	0.120

그림 4-3. 수집된 절삭성 데이터베이스의 face milling의 경우에 대한 예

표 4-4. 가공작업의 설계 또는 실행상의 고려사항 항목과 응답내용

순서	구분	고려사항	응답내용 (해당란에 빈도)		
			항상 고려한다.	가끔 고려한다.	전혀 고려치 않는다.
1	공작기계 관련된 사항	1 공작기계의 정밀도는 만족되는가?	2	7	1
		2 공작기계 스톨들의 강성은 충분한가?	1	6	3
		3 공작기계의 동력은 이 가공에 충분한가?	3	6	1
		4 공작기계가 가공중 진동을 일으킬까?	1	5	4
		기타의 사항이 있으면 기입해 주십시오. * 공작기계의 동시제어축 수. ( 동시1, 2축 기계도 가끔 있음) * 축 제어시의 한계 부하 ( 대형 부품의 가공시 )			
2	被削材의 取付 에 관한 사항	1 지그 및 고정구의 강성은 충분한가?	6	4	0
		2 커터의 운동이나 칩 배출에 방해되지 않을까?	5	4	1
		3 取付방법이 가공방향으로 되어 있는가?	3	4	3
		4 피삭재의 교환(착탈)은 용이한가?	2	4	4
		5 取付具가 진동을 일으키지는 않을까?	4	4	2
		6 切削熱에 의해 변형을 일으키지는 않을까?	4	5	1
		7 절삭 칩이 한 군데에 물리지는 않을까?	5	5	0
		기타의 사항이 있으면 기입해 주십시오. * 작업의 안전과 직결되므로 고려함. * 가끔 잘못 판단으로 미스가 생기기도 하므로 주의하려 노력함. * 공정설계에서 경험상 중요하다고 생각함.			
3	절삭공구에 관련된 사항	1 절삭공구의 강성은 충분한가?	5	5	0
		2 커터의 형상은 피삭재의 가공에 적합한가?	3	5	2
		3 커터의 수명(가공목표시간)은 얼마가 될까?	4	4	2
		4 커터의 수명판정기준은 무엇으로 할까?	3	4	3
		5 절삭열이 과다하거나 불꽃이 일어나지 않을까?	4	6	0
		6 절삭날에 결손이나 파손의 위험은 없는가?	5	4	1

<< 다음 편에 계속됨 >>

----- ( 계속 ) -----

순 서	구 분	고 려 사 항	응 답 내 용 (해당난에 빈도)		
			항상 고려한다.	가끔 고려한다.	전혀 고려 치 않는다.
3	---- 계속 ---	7	4	5	1
		8	2	5	3
		9	3	6	1
		기타의 사항이 있으면 기입해 주십시 요.	<ul style="list-style-type: none"> <li>* 우선 절삭날에 결함이 없는가 점검.</li> <li>* 해당공구로 끝까지 가공할 수 있는가를 판단한 후에 사용함.</li> <li>* 이들 사항은 가공 초기에 가공상태를 지켜보면서 판단함.</li> <li>* 가급적 안전하고 여유있게 함.</li> <li>* 커터가 깊어서 처짐이 많을까 생각함.</li> </ul>		
4	가공 작업에의 관련 및 기타	1	8	2	0
		2	9	1	0
		3	8	2	0
		4	7	3	0
		5	2	5	3
		6	4	4	2
		7	4	4	2
		8	7	3	0
		9	8	2	0
		0	2	7	1
		1	6	3	1
		2	3	5	2
		기타의 사항이 있으면 기입해 주십시 요.	<ul style="list-style-type: none"> <li>* 절삭유는 한, 두가지를 대강사용함.</li> <li>* 8,9항은 작업의 안전상 항시고려함.</li> <li>* 표면거칠기는 가끔 계산해 보거나 표류 찾아보기도 함.</li> <li>* 칩배출은 드릴링에서만 주로 점검함.</li> <li>* 1,2,3항은 높이고 싶으나 사고의 우려가 있어서 안전하게 놓음.</li> </ul>		



## 제 2 절 DB editor 의 프로그래밍

### 1. Multi-Layer 퍼셉트론의 구성

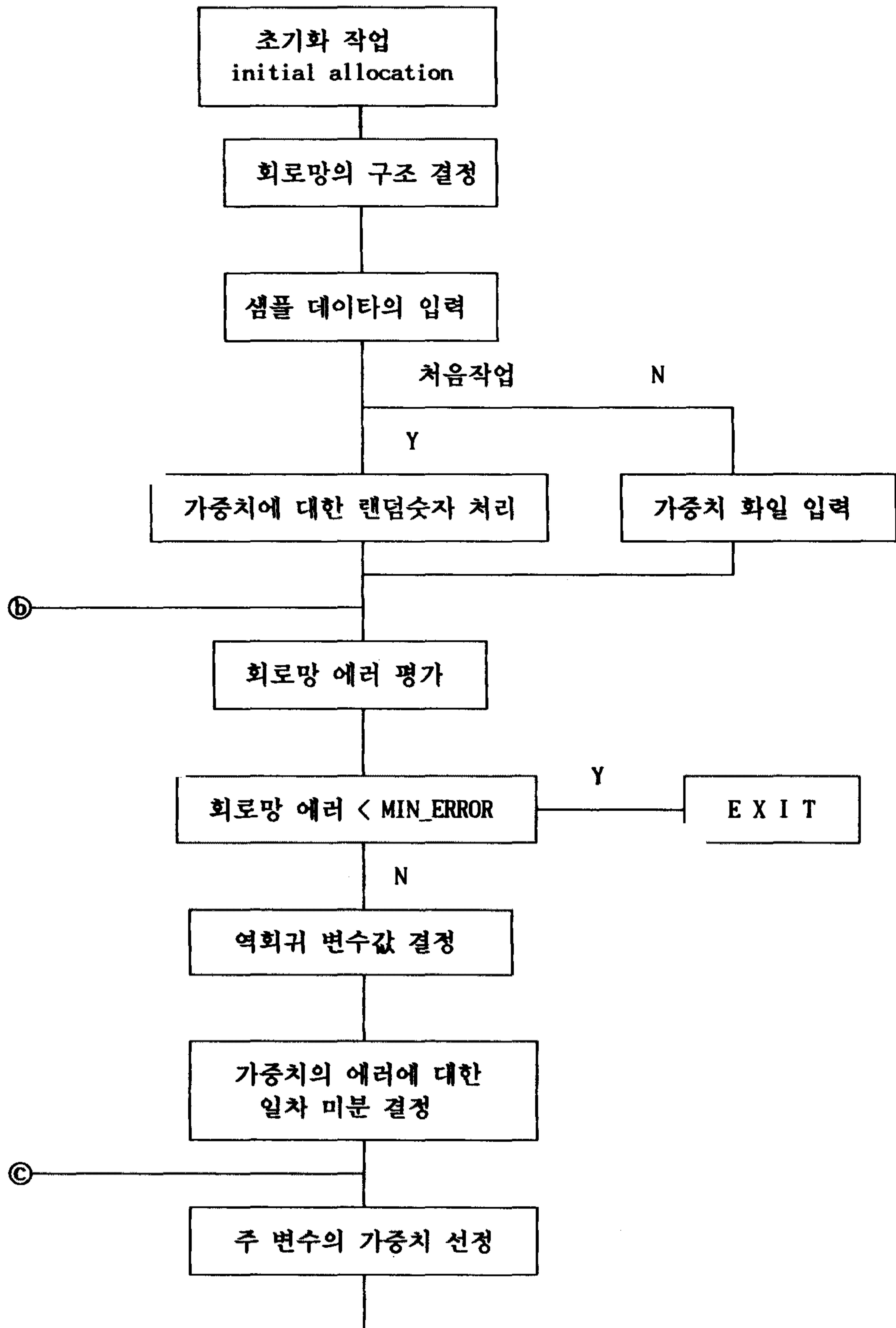
앞의 3-2절에서의 매핑관계를 구성하는 방법에 있어서 다층 퍼셉트론 방법은 숨은층 (hidden layer)의 수행과정에서 비교적 잘 수렴되었으며, 메모리의 부족현상이 거의 생기지 않았고, 임의 절삭계수를 연속적으로 선정할 수 있기 때문에 이산화된 절삭계수가 아닌 가공조건에 맞는 최적의 절삭계수를 얻을 수 있었을 뿐 아니라, 최적화 과정에 있어서도 local minimum 현상이 거의 생기지 않았다.

따라서 본 연구에서는 다층퍼셉트론을 통하여 신경회로망의 소자를 구성하였고 이를 학습절차를 통해 수학적 모형화 결과의 정확성을 판별케 하여 이를 최소화하는 방법을 사용하기로 하였는데, 그림 4-4는 이 다층 퍼셉트론 소자를 구성하는 절차를 나타낸 flow diagram 이다.

초기화 작업을 수행한 이후에는 회로망의 구조가 결정되며, 학습을 위한 샘플 데이터가 입력된다. 신경회로망의 가중치는 초기에 임의 숫자를 사용하여 입력되며, 이러한 가중치와 신경회로망의 구조 변수를 사용한 출력이 검사되어 주어진 실제 결과와의 에러를 결정한다. 이러한 에러에 대하여 델타 규칙과 역전파에 의하여 전체 에러에 대한 각 가중치의 구배를 구하게 되고, 초대 감소 경사의 방향으로 가중치를 변화시키게 된다.

이러한 과정을 반복하여 에러가 기준값 이하로 작아지면 반복 학습 과정을 중단하고 신경회로망의 가중치를 결정하게 되는 것이다.

이때 각 가중치 구배에 의한 가중치 변화량은 황금 분할법에 의한 에러 곡면의 탐색과 수렴을 안정되게 하기 위한 휴리스틱을 적용하여 결정한다.





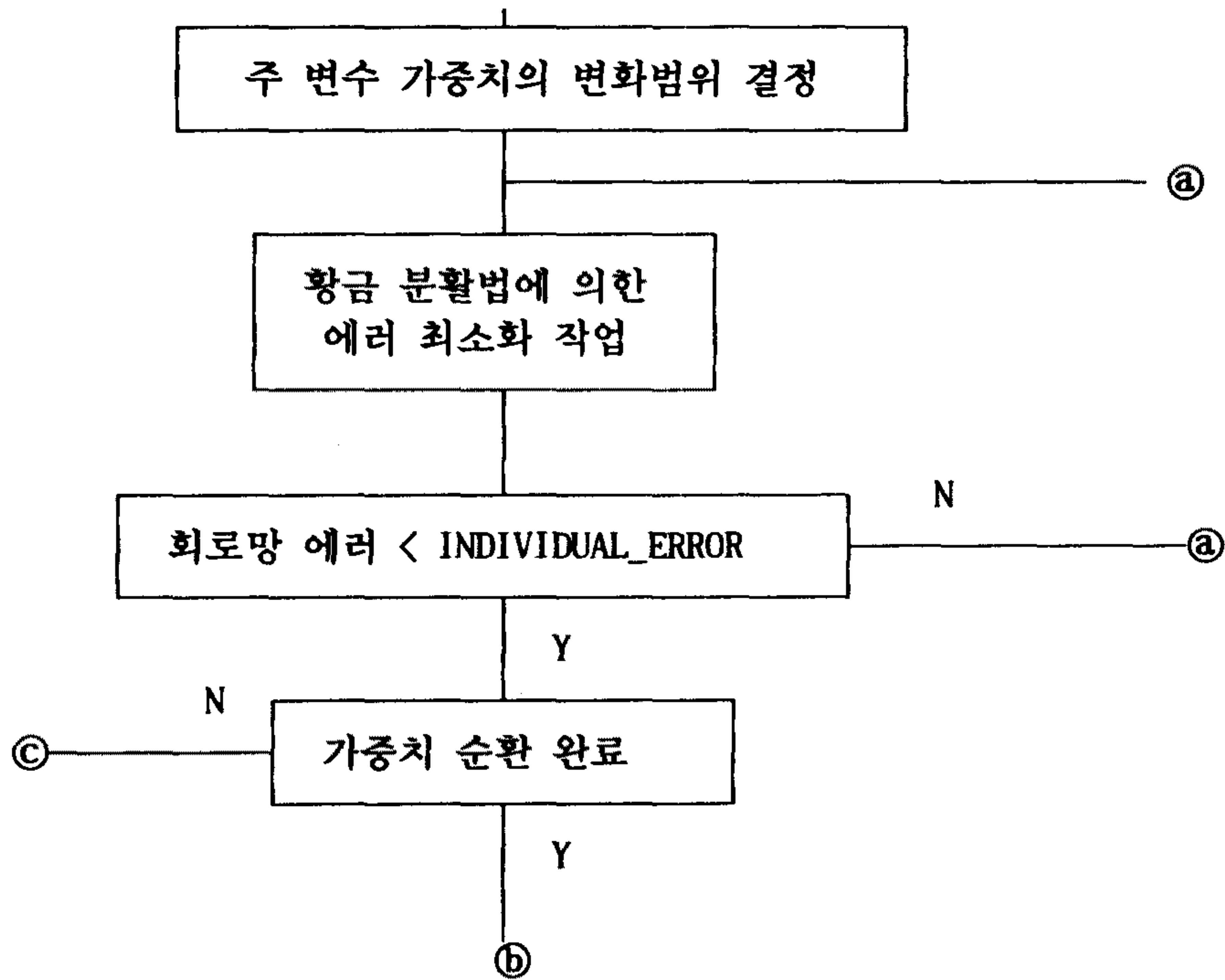


그림 4-4. 다층 퍼셉트론의 구성을 위한 flow diagram

신경회로망을 사용한 데이터 출력의 경우에는 그림 4-5와 같은 과정을 거치게 된다. 특징 클래스, 데이터 제공 그룹에 의해 신경회로망 데이터 파일이 결정되게 한 후에는 회로망 층수, 각 층의 노드 수, 입력 데이터의 평균 및 스케일링 값, 출력 데이터의 평균 및 스케일링 값 그리고 퍼셉트론의  $\beta$ ,  $\theta$ , threshold, 각 퍼셉트론의 연결 가중치값을 결정한다. 입력 및 출력 과정에서는 스케일링을 행한후 신경회로망을 전파하게 된다. 퍼셉트론의 전달함수는 시그모이드 함수를 사용하였다.

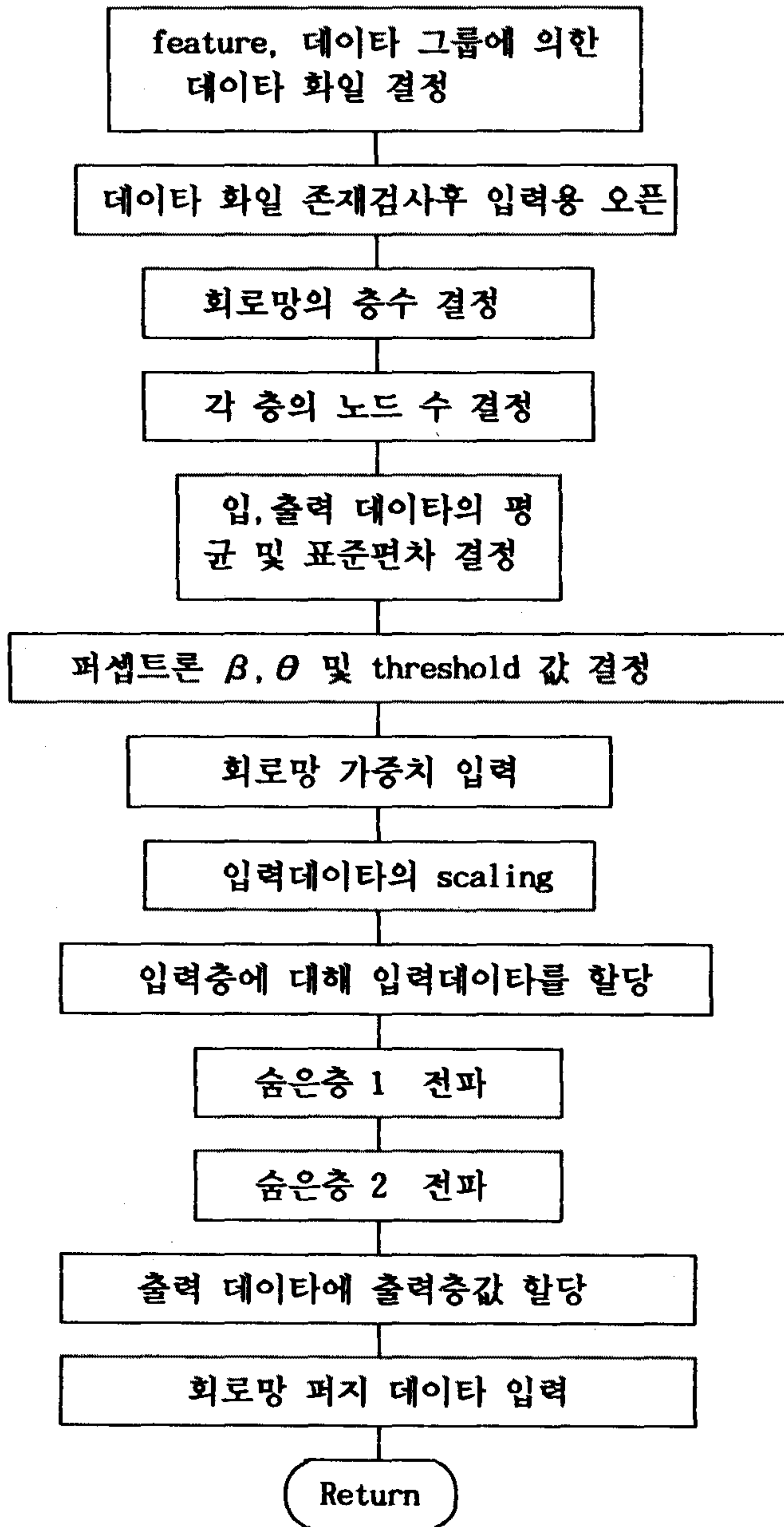
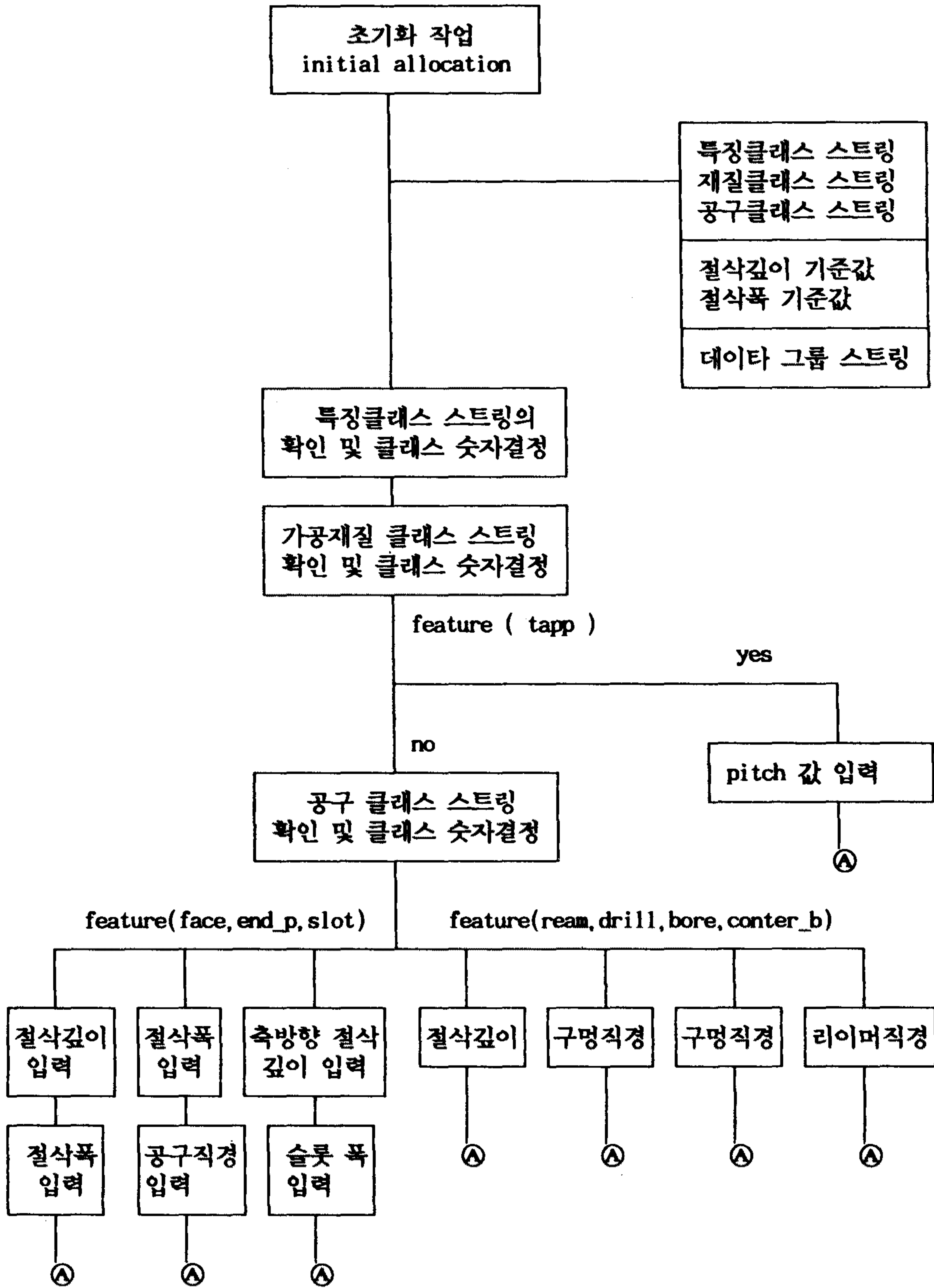


그림 4-5. 숨은층 네트워크 전파과정의 flow diagram

그림 4-6은 장차 main module에서 절삭속도와 이송속도를 결정할 때 사용되는, 입력 데이터의 퍼지 정보, 회로망의 퍼지 가능성 값을 결정하는 과정을 도시한 것이다.

신경회로망의 학습 데이터 구성과 마찬가지로 초기화 작업이 수행된 후에 특징 클래스의 스트링에 대한 패턴 연결과 재료 클래스의 스트링에 대한 패턴 연결을 구성한다. 그 후 각 특징 클래스에 따라 입력 데이터의 갯수와 값을 결정하게 된다. 이러한 입력 데이터에 대하여 입력 데이터 허용범위를 검토하고 본 연구과정의 신경회로망 학습을 위하여 5개의 그룹에서 제공되었던 5개의 절삭 속도에 대하여 기준값을 중심으로 각 입력 값과 신경회로망 자체에 대한 퍼지 가능성 값을 결정한다. 이를 바탕으로 각 절삭속도와 퍼지 가능성 값을 사용하여, 센트로이드 값을 취함으로써 최종적인 절삭 속도를 결정하게 된다.

그리고 그림 4-7에서와 같이 입력 데이터는 여러개의 학습 기준값과 그 퍼지 가능성으로 매핑이 되어 신경회로망을 통과하게 되는데, 이때에는 입력값에 가장 가까운 좌우의 기준값을 사용한다. 이런 매핑의 결과로 보링, 리밍, 드릴링, 카운터 보링, 태핑의 클래스의 경우에는 2쌍의 입력 데이터와 퍼지 값이 결정되며, 페이스 밀링, 엔드 밀링 윤곽 절삭, 슬로팅의 클래스의 경우에는 4쌍의 입력 데이터와 퍼지 값이 결정된다.



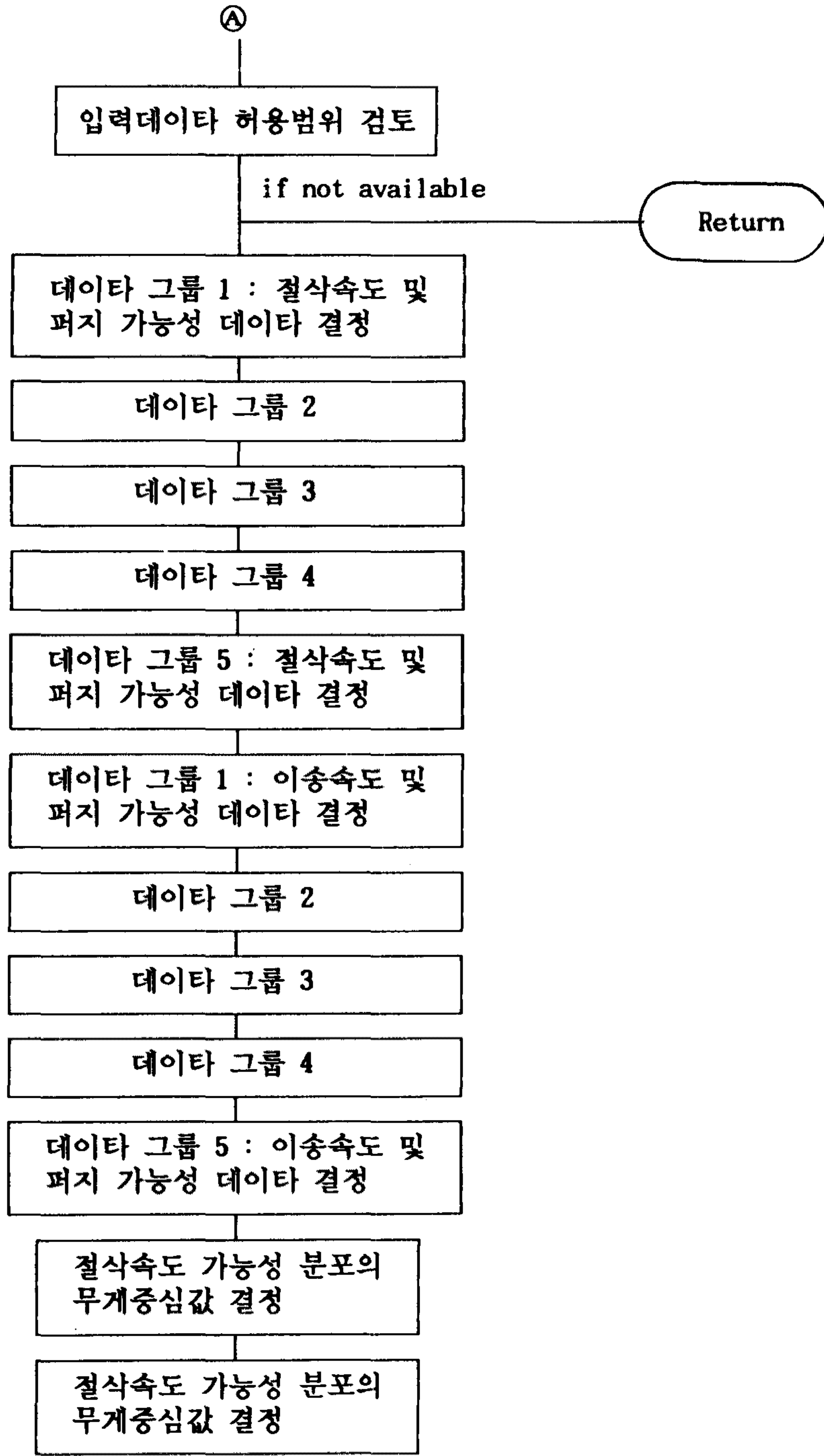
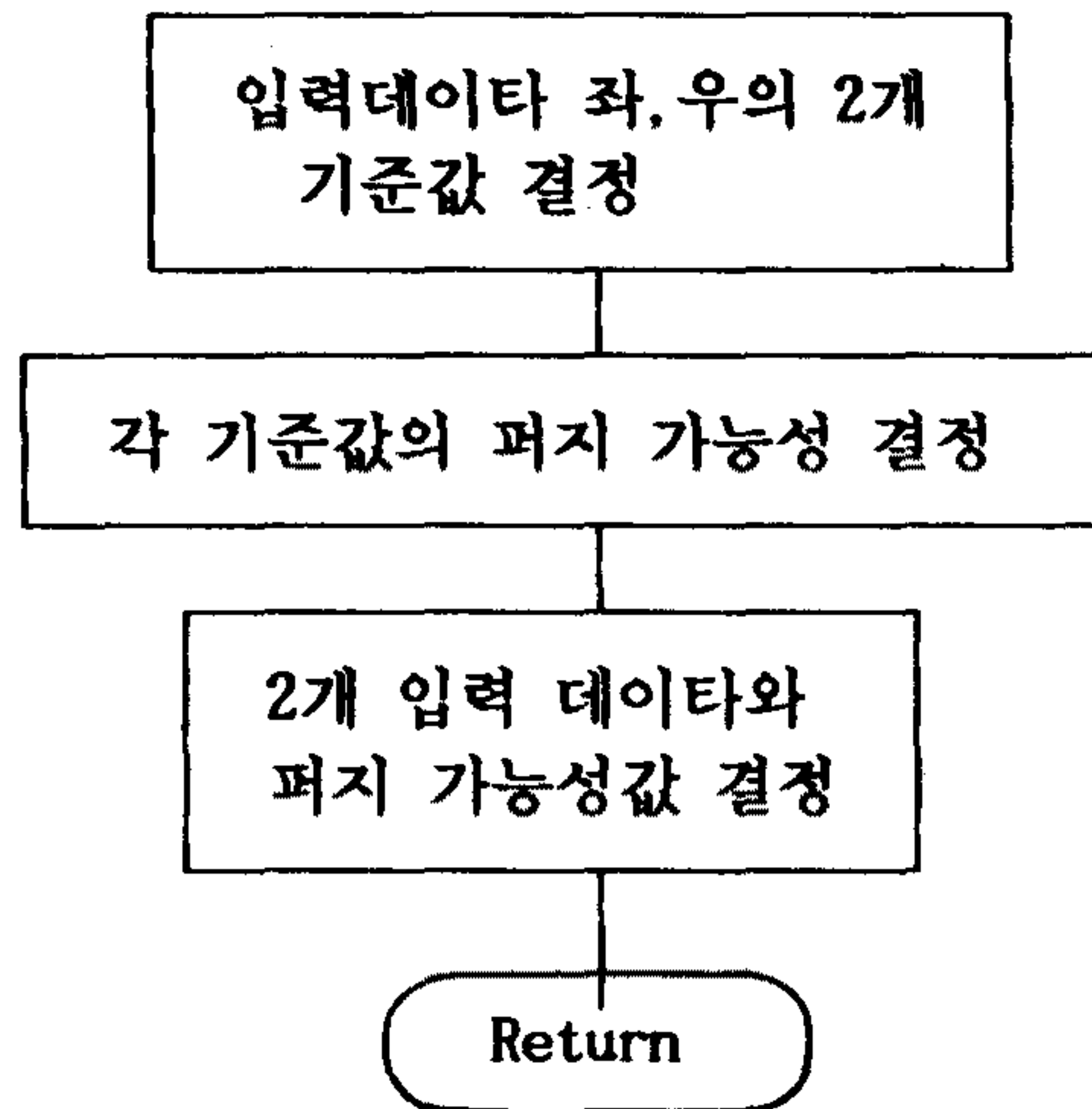


그림 4-6. 퍼지 정보 처리를 위한 flow diagram

[ Feature ( bore, drill, Count\_b, ream, tapp ) ]



[ Feature ( face, end\_p, slot ) ]

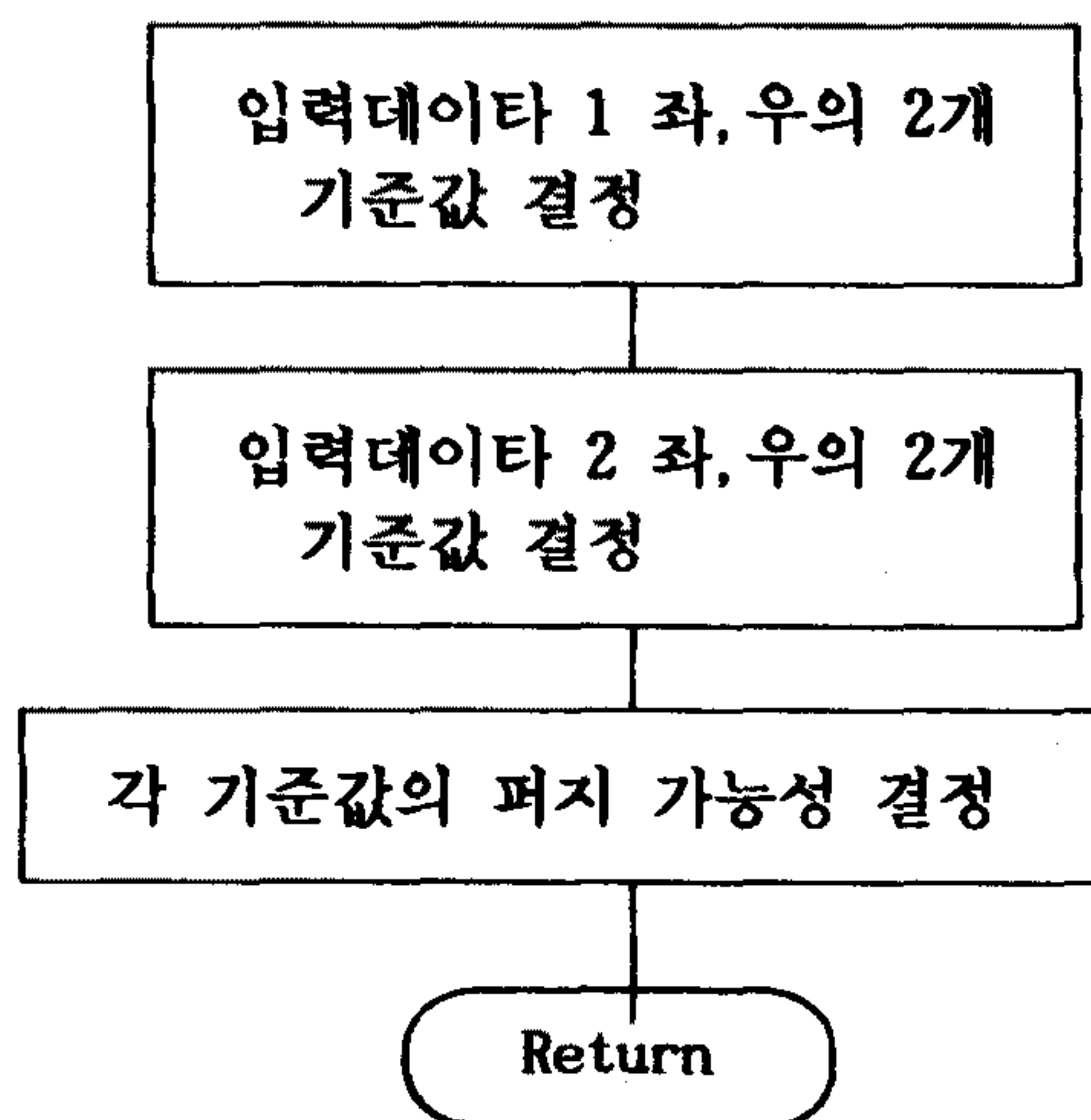


그림 4-7. 네트워크 landmarks에의 입력 매핑에 대한 flow diagram

## 2. 신경 회로망 데이터 입력 파일 구조

신경회로망 가중치 데이터 파일의 파일명은 특징 그룹의 구분자 (feature identifier), 데이터 작성그룹 구분자 (data group identifier), 절삭속도 및 이송속도 간의 구분자로 구성되어 있다. 예를 들어 페이스 밀링 경우에 A 그룹에서 제공한 경험데이터로 이송속도를 위한 신경회로망을 구성하였을 때에는 그 회로망의 구조 및 가중치 등이 "face", "\_a", "\_f"의 문자열이 합쳐진 face\_a\_f.wgt의 파일에 저장된다.

파일은 순차적 접근방식의 일반적인 파일로 저장되어 있으며, 입력속도의 문제는 중요하지 않으므로 관리의 용이성과 차후에 학습 데이터의 보충으로 인하여 재학습되어 개선된 데이터를 저장·관리가 용이하도록 하기 위하여 ASCII 코드의 일반 파일로 저장한다.

입력되어 있는 데이터는 학습과정의 성능 지수 및 학습과정에 대한 설명이 있게 되며, 회로망의 층수, 회로망 각 층의 노드수, 학습 입력 데이터의 평균 값, 학습 입력 데이터의 스케일링 값, 학습 출력 데이터의 평균값, 학습 출력 데이터의 스케일링 값, 신경회로망 전체 퍼셉트론의 입력 기준값 (threshold),  $\beta$  값과  $\theta$  값이 저장되어 있으며, 각 가중치의 회로상의 인덱스와 가중치 값이 저장되어 있다.

가중치 파일의 구조는 그림 4-8에 나타나 있으며, 파일의 예는 다음과 같다.



[ 가중치 파일의 예 ]

```
current error : 111.181504
SUCCESS in LEARNING WEIGHT at 3 iteration

num_layer: 4
num_node: 3 7 7 1
average: 15.689655 2.500000 2.500000 0.062278
scale: 18.603449 1.950000 1.950000 0.075761
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: -2.845333
weight[1][0][1]: -7.653771
weight[1][0][2]: -1.718533
weight[1][1][0]: -6.098242
weight[1][1][1]: 2.193134
weight[1][1][2]: 12.498396
weight[1][2][0]: 7.666320
weight[1][2][1]: -10.609221
weight[1][2][2]: 3.808480
weight[1][3][0]: 2.917535
weight[1][3][1]: -17.976295
weight[1][3][2]: 4.481180
weight[1][4][0]: -1.152853
weight[1][4][1]: -7.315074
weight[1][4][2]: -1.281712
weight[1][5][0]: -5.283306
weight[1][5][1]: -2.956119
weight[1][5][2]: -6.769426
weight[1][6][0]: 10.807340
weight[1][6][1]: 5.486214
weight[1][6][2]: 6.211746
weight[2][0][0]: 3.993208
weight[2][0][1]: 4.660013
weight[2][0][2]: 4.718709
weight[2][0][3]: 6.914861
weight[2][0][4]: -8.377044
weight[2][0][5]: 11.060924
weight[2][0][6]: -3.179402
weight[2][1][0]: -6.985947
weight[2][1][1]: -6.879461
weight[2][1][2]: -4.986746
weight[2][1][3]: -7.226872
weight[2][1][4]: -7.013579
```

```
weight[2][1][5]: 4.127073
weight[2][1][6]: -5.152562
weight[2][2][0]: -4.226021
weight[2][2][1]: 2.178013
weight[2][2][2]: -3.617895
weight[2][2][3]: -1.087657
weight[2][2][4]: -1.534396
weight[2][2][5]: 2.840471
weight[2][2][6]: -6.167995
weight[2][3][0]: 6.477601
weight[2][3][1]: -2.801642
weight[2][3][2]: -5.056403
weight[2][3][3]: -1.796334
weight[2][3][4]: 8.799793
weight[2][3][5]: -7.487347
weight[2][3][6]: 2.271102
weight[2][4][0]: -8.070490
weight[2][4][1]: 7.384803
weight[2][4][2]: -4.679778
weight[2][4][3]: 6.021894
weight[2][4][4]: 8.341533
weight[2][4][5]: 5.577064
weight[2][4][6]: 0.530860
weight[2][5][0]: 6.654666
weight[2][5][1]: 3.554341
weight[2][5][2]: -4.785960
weight[2][5][3]: 8.611599
weight[2][5][4]: 6.329325
weight[2][5][5]: -5.545141
weight[2][5][6]: 1.633716
weight[2][6][0]: -0.799187
weight[2][6][1]: -1.714016
weight[2][6][2]: -3.236910
weight[2][6][3]: -0.205491
weight[2][6][4]: 4.459275
weight[2][6][5]: 5.153455
weight[2][6][6]: -3.643570
weight[3][0][0]: 3.063407
weight[3][0][1]: 5.497949
weight[3][0][2]: -5.378568
weight[3][0][3]: -7.365783
weight[3][0][4]: -1.382810
weight[3][0][5]: 8.539319
weight[3][0][6]: -8.468186
network_fuzzy_number: 0.460000
```

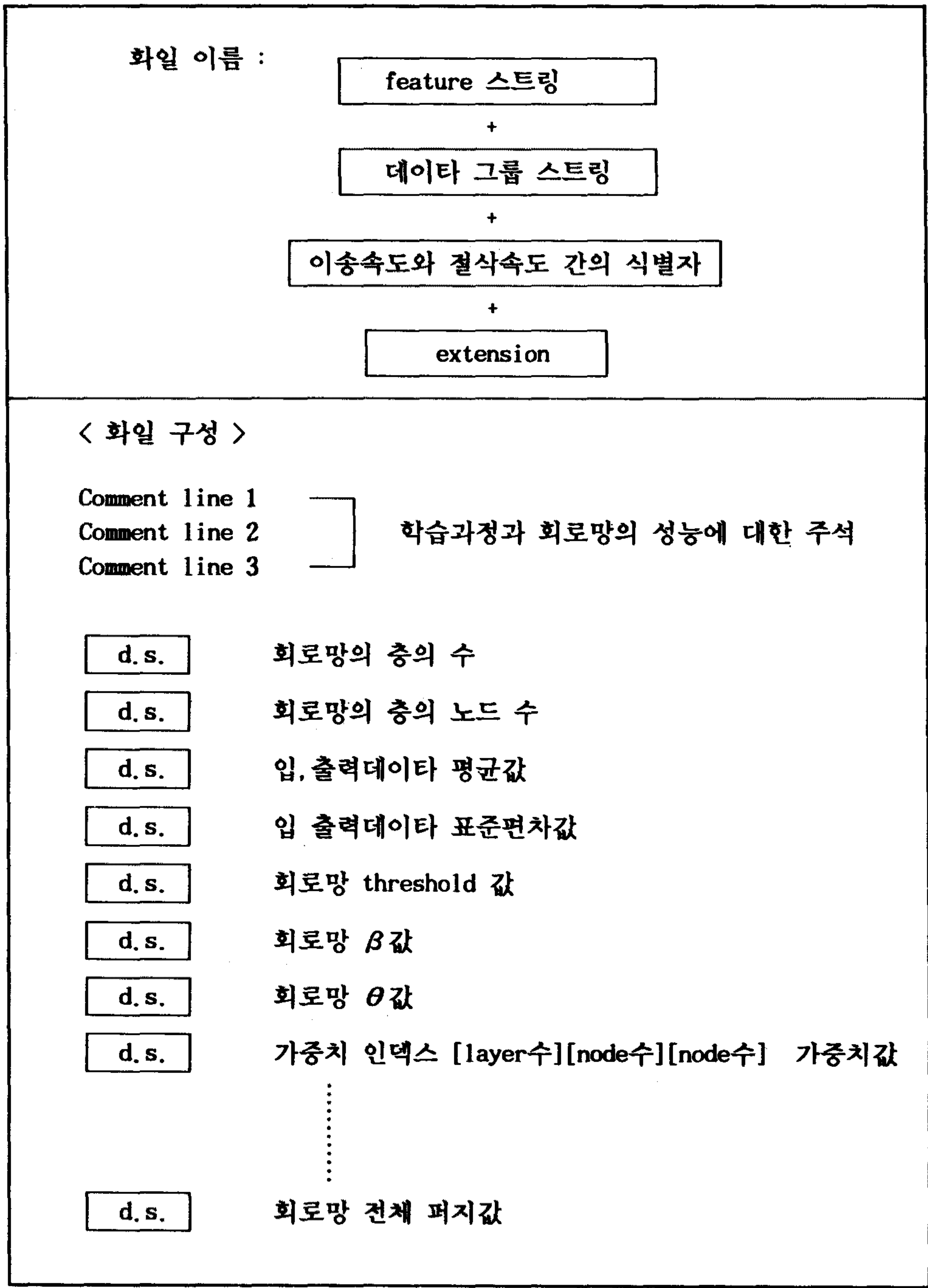


그림 4-8. 가중치 파일의 구조

### 3. 신경회로망의 학습과정

신경회로망의 숨은층의 층수와 각 층의 노드 수, 그리고 각 퍼셉트론의 전달함수(sigmoid function)는 학습과정 초기에 고정되어 시도될 수도 있으며, 학습을 효율적으로 하기 위해 신경회로망의 구조를 동적으로 효율적으로 변환하는 것에 대한 연구도 활발히 진행되고 있다.

본 프로그래밍에서는 신경회로망의 구조와 퍼셉트론의 구조를 고정하고 신경회로망의 가중치만 변화시키는 방식을 취하였다.

초기의 가중치는 원형(prototype of weights)을 사용하여 학습이 빨리 되도록 하는 연구 결과도 있으나, 아직 확립된 방법이 아니므로 일반적으로 -1.0에서 1.0 사이의 임의 숫자 처리방식으로 초기값을 결정하였다.

학습 과정은 델타 규칙과 역전파의 방법을 사용하여 구성하였다. 페이스 밀링과 엔드 밀링과 슬로팅에서는 하나의 신경회로망을 위한 샘플 데이터의 갯수가 약 2000개 정도되므로 에러 곡면이 기복이 심하고 델타 규칙에 의하여 구해진 변화 방향 벡터에 대하여 아주 민감하게 변화하게 되는 특성을 지니고 있었다.

따라서 에러 곡면에서 가중치 변수의 전체 벡터량에 의한 탐색보다 각 가중치 하나하나에 대하여 에러 곡면을 탐색하는 것이 최소 에러를 찾는 데 안정성을 제공하므로 이러한 방법을 적용하였다.

## 제 3 절 절삭가공 데이터 베이스의 구축

### 1. 데이터 베이스 파일 시스템

각 특징 클래스의 신경회로망이 구성되어 파일 시스템으로 저장되는 과정은 다음의 그림 4-9와 같다. 각 데이터 그룹별로 그리고 특징 클래스별로 저장되어 있는 학습 데이터 파일로부터 학습 데이터를 읽어 들인후 유용한 학습 방법을 선택하여 학습을 실행한후 절삭 속도와 이송속도별로 파일에 저장되게 된다.

이 파일에 저장된 데이터는 그 관계가 선명하므로 질의를 효율적으로 처리할 필요가 차후 생기게 되는 경우에는, 데이터의 관계 테이블이 쉽게 결정되므로 관계형 데이터 베이스 관리 시스템으로의 전환이 간단한 로딩 모듈에 의하여 구성될 수 있다.

### 2. Fuzzy 값 처리에 의한 데이터 액세스

Fuzzy값 처리에 의한 데이터 액세스는 다음과 같은 과정을 거치게 된다.

① 입력 데이터 값이 실수로 주어지면 절삭 조건 선정을 위한 신경회로망의 학습에 사용된 학습 데이터의 기준값의 순서 집합에 대하여 그 순서를 결정하여 입력 데이터의 좌·우에 있는 기준값을 신경회로망 입력 값으로 사용한다. 즉 하나의 입력변수값에 대하여 2개의 기준값이 사용되게 된다.

② 입력 데이터 값과 신경회로망 입력 기준값의 차이는 신경회로망 입력값의 퍼지 가능성 값으로 전환된다. 즉 하나의 입력변수가 두개의 기준점으로 매핑되는 대신에 그 기준점은 적절한 퍼지 값을 갖게 되는 것이다.

그리고 퍼지 가능성 값은 다음의 식을 사용하는 것으로 하였다.

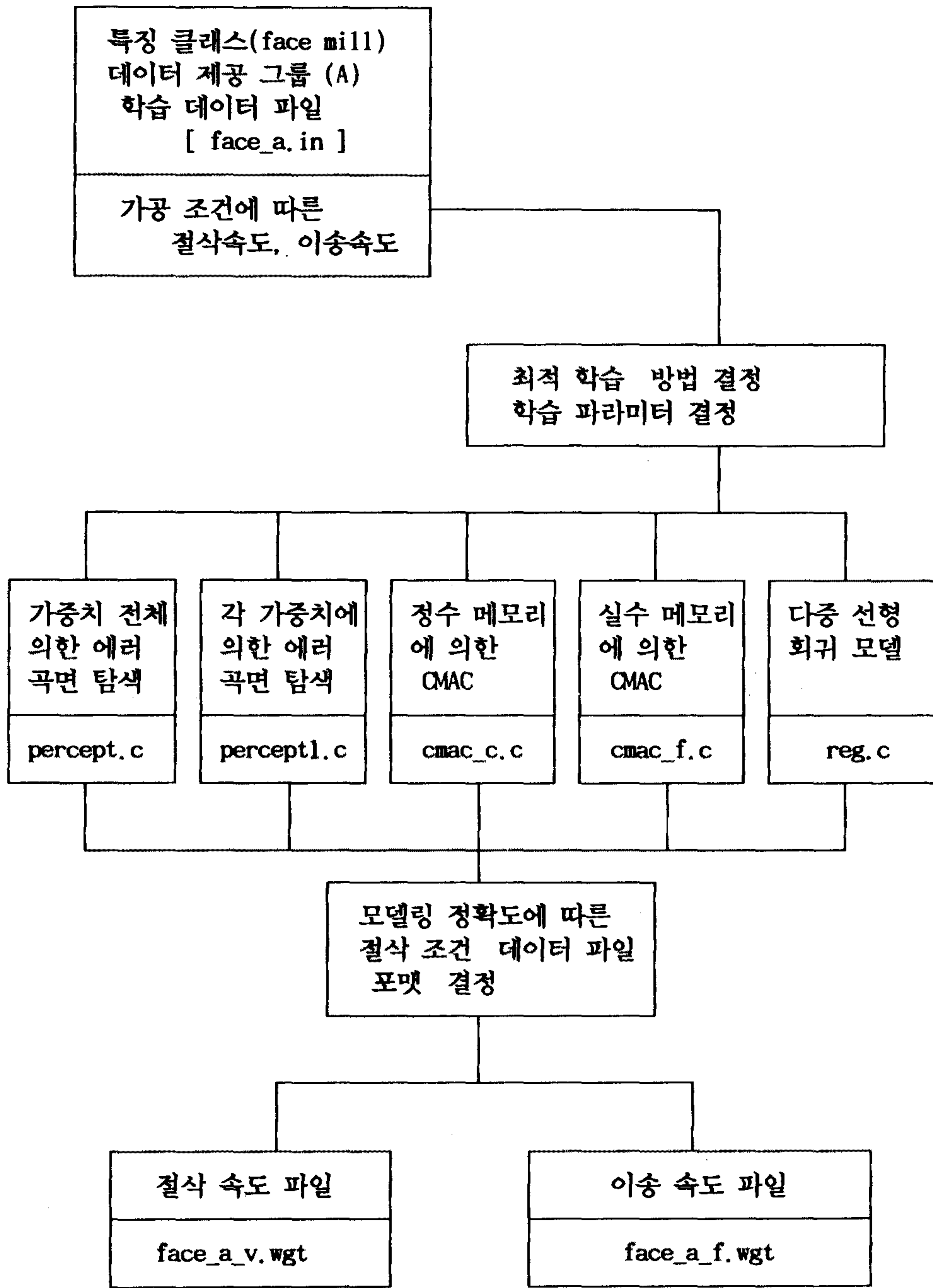


그림 4-9. 절삭조건 결정용 데이터파일의 구축과정



좌측의 기준값은

$$\text{fuzzy\_left} = F\_MIN + (1.0 - F\_MIN) * (\text{input\_data\_value} - \text{left\_landmark}) / (\text{right\_landmark} - \text{left\_landmark}).$$

우측의 기준값은

$$\text{fuzzy\_right} = F\_MIN + (1.0 - F\_MIN) * (\text{right\_landmark} - \text{input\_data\_value}) / (\text{right\_landmark} - \text{left\_landmark})$$

으로 주어진다. 여기서 하나의 경험값으로 사용되는 F\_MIN의 값은 퍼지 가능성 값의 최소값을 정의하고 좌우 기준점의 퍼지 값에 영향을 미치는 파라미터이다.

③ 해당 특징 클래스 및 데이터 그룹의 신경회로망 파일로부터 신경회로망 구조와 변수 파라미터, 가중치 값을 읽어 들인 후 앞에서 구해진 기준값들의 쌍을 입력으로 하여 신경회로망을 전파시켜 출력을 얻게 된다.

④ 출력 값의 퍼지 가능성 값은 퍼지집합의 확장 원리(extension principle)을 이용하여 결정하게 되는데, Mamdani의 최소-최대 연산을 사용하여 결정한다.

⑤ 각 데이터 그룹으로부터 얻어진 입력 기준점들에 의하여 신경회로망으로부터 출력된 절삭 속도와 그 퍼지 값에 대하여 탈 퍼지화(defuzzification)에 의하여 센트로이드 값을 결정하여 최종적인 절삭속도로 결정한다.

마찬가지로 각 데이터 그룹으로부터 얻어진 입력 기준점들에 의하여 출력된 이송 속도와 그 퍼지 값에 대하여서도 탈 퍼지화(defuzzification)에 의하여 센트로이드 값을 결정하여 최종적인 이송속도로 결정한다.

### 3. 규칙에 의한 데이터의 갱신

앞의 과정에 의해서 구축된 5 group의 절삭계수 데이터베이스는 절삭조건 결정 프로그램에 의해서 입력된 가공제어인자 파라미터들에 의해서 절삭조건이 결정되는데, 이 과정은 우선 5그룹에서 각각 액세스되는 절삭계수들을 fuzzy값처리에 의해서 결정되게 되고, 이렇게 결정된 절삭조건은 다시 각 feature별로 주어진 modify rule에 의해서 설정되는 correction (또는 reduction) factor로서 그 값이 갱신되는 과정을 거친다. 이들 과정에서 처리되는 주요 갱신 룰들을 각 feature 별로 살펴보면 다음과 같다.

#### (1) Face Milling

① Finishing allowance (다듬질 가공인 경우의 표면거칠기 제한):

$$\text{IF } Y2 > f_{sh} \quad \text{THEN } \overline{Y2} = f_{sh}$$

$$\text{여기서 } f_{sh} = 2 \cdot \sqrt{R_{nose} \cdot R_m}$$

$R_{nose}$  : 커터 인서트의 노오즈 반경 ( mm )

$R_m$  : 표면거칠기의 상한치 ( mm )

② Power restriction ( 최대절삭동력제한 ) :

$$\text{IF } P_{kw} > P_{Nominal} \quad \text{THEN } Y2 = Y2 - ( Y2 - f_{kw} ) / 2$$

$$\text{AND } Y1 = Y1 \cdot ( f_{kw} / Y2 )$$

$$\text{여기서 } P_{kw} = \frac{k_s \cdot X3 \cdot X4 \cdot Y2 \cdot Z_n \cdot N_r}{6120000 \cdot \eta}$$

그리고  $k_s$  : 비절삭저항값 ( kg/mm<sup>2</sup> )

$\eta$  : 기계효율 ( 0.7-0.8, 0.7 기준으로 사용 )

$N_r$  : 스피들 회전수 (  $N_r = ( 1000 Y1 ) / ( \pi D )$  )

$P_{Nominal}$  : 공작기계의 허용동력 ( kW )

$Z_n$  : 커터의 날수 ( teeth/ rev. )

또한

$$f_{kW} = \frac{6120000 \cdot \eta}{k_s \cdot X_3 \cdot X_4 \cdot Z_n \cdot N_r} P_{Nominal}$$

표 4-5.  $X_1$  값 ( 재질 class 값 ) 에 따른  $k_s$  의 공칭값.

$X_2$ ( 재질 class )	$k_s$ ( kg/mm <sup>2</sup> )
1, 2, 3, 4, 5, 6, 7	250
8, 9, 10, 11, 13	240
12, 14, 15, 16, 17	210
18, 19, 20, 21	190
22, 23, 24, 25, 26	170
27, 28, 29, 30	130

③ 축진동 발생 가능성 :

IF  $Y_2 < 0.15$  AND  $N_r > 2000$

THEN  $Y_1 = 2 \pi D$

④ 최대 스피들속도 고정 :

IF  $N_r > N_{allow}$  THEN  $Y_1 = \frac{\pi D}{1000} N_{allow}$

여기서  $N_{allow}$  : 허용최대 스피들 속도

⑤ 주변력 ( peripheral force ) 의 제한 :

$$\text{IF } F_p > F_{allow} \quad \text{THEN} \quad Y_2 = \frac{2 \cdot D \cdot F_{allow}}{k_s \cdot Z_n \cdot X_3 \cdot X_4}$$

여기서  $F_p = k_s \cdot (Z_n/2) \cdot X_3 \cdot Y_2 \cdot (X_4/D)$

$F_{allow}$  : 최대허용 주변력 ( kg )

⑥ 편심력 크기의 제한 :

$$\text{IF } M_p > M_{allow} \quad \text{THEN} \quad Y_2 = \frac{4 D \cdot M_{allow}}{k_s \cdot Z_n \cdot X_3 \cdot X_4 ( D - X_4 )}$$

여기서  $M_p = F_p ( D - X_4 ) / 2$

$M_{allow}$  : 최대허용 편심원 모멘트 ( kg-mm )

⑦ 칩발생 불완전요소의 제한 :

$$\text{IF } ( X_3/Y_2 > 30 \text{ OR } X_3/Y_2 < 0.5 ) \text{ AND } N_r > 2000$$

$$\text{THEN } Y_1 = 2 \pi D$$

(2) End Milling - Peripheral

① Finishing allowance :

$$\text{IF } Y_2 > f_{sh} \quad \text{THEN} \quad \overline{Y_2} = f_{sh}$$

여기서  $f_{sh} = \sqrt{2 \cdot X_4 \cdot R_m}$

$R_m$  : 표면거칠기의 상한치 ( mm )

② 최대절삭동력제한 :

$$\text{IF } P_{kw} > P_{Nominal} \quad \text{THEN} \quad Y_2 = Y_2 - ( Y_2 - f_{kw} ) / 2$$

$$\text{AND } Y_1 = Y_1 \cdot ( f_{kw} / Y_2 )$$

여기서  $P_{kw} = \frac{k_s \cdot X_3 \cdot X_4 \cdot Y_2 \cdot Z_n \cdot N_r}{4080000 \cdot \eta}$

그리고  $k_s$  : 비절삭저항값 (  $\text{kg}/\text{mm}^2$  )

$\eta$  : 기계효율 ( 0.6-0.8, 0.6 기준으로 사용 )

$N_r$  : 스피들 회전수 (  $N_r = ( 1000 Y_1 ) / ( \pi X_4 )$  )

$P_{\text{Nominal}}$  : 공작기계의 허용동력 ( kW )

$Z_n$  : 커터의 날수 ( teeth/ rev. )

또한

$$f_{kW} = \frac{4080000 \cdot \eta}{k_s \cdot X_3 \cdot X_4 \cdot Z_n \cdot N_r} P_{\text{Nominal}}$$

③ 축진동 발생 가능성 :

IF  $Y_2 < 0.025$  AND  $N_r > 3000$

THEN  $Y_1 = 3 \pi X_4$

④ 최대 스피들속도 고정 :

$$\text{IF } N_r > N_{\text{allow}} \quad \text{THEN } Y_1 = \frac{\pi X_4}{1000} N_{\text{allow}}$$

여기서  $N_{\text{allow}}$  : 허용최대 스피들 속도

⑤ 주변력 ( peripheral force ) 의 제한 :

$$\text{IF } F_p > F_{\text{allow}} \quad \text{THEN } Y_2 = \frac{F_{\text{allow}}}{0.75 k_s \cdot Z_n \cdot X_3}$$

여기서  $F_p = 0.75 k_s \cdot Z_n \cdot X_3 \cdot Y_2$

$F_{\text{allow}}$  : 최대허용 주변력 ( kg )

⑥ 칩발생 불안전요소의 제한 :

IF (  $X_3/Y_2 > 20$  OR  $X_4/Y_2 < 20$  ) AND  $N_r > 2000$

THEN  $Y_1 = 2 \pi X_4$

### (3) End Milling - Slotting & Pocketing

#### ① Finishing allowance :

$$\text{IF } Y2 > f_{sh} \quad \text{THEN } \overline{Y2} = f_{sh}$$

$$\text{여기서 } f_{sh} = \sqrt{2 \cdot X4 \cdot R_m}$$

$R_m$  : 표면거칠기의 상한치 ( mm )

#### ② 최대절삭동력제한 :

$$\text{IF } P_{kw} > P_{Nominal} \quad \text{THEN } Y2 = Y2 - ( Y2 - f_{kw} ) / 2$$

$$\text{AND } Y1 = Y1 \cdot ( f_{kw} / Y2 )$$

$$\text{여기서 } P_{kw} = \frac{k_s \cdot (X4)^2 \cdot Y2 \cdot Z_n \cdot N_r}{4080000 \cdot \eta}$$

그리고  $k_s$  : 비절삭저항값 (  $\text{kg/mm}^2$  )

$\eta$  : 기계효율 ( 0.6-0.8, 0.6 기준으로 사용 )

$N_r$  : 스피들 회전수 (  $N_r = ( 1000 Y1 ) / ( \pi X4 )$  )

$P_{Nominal}$  : 공작기계의 허용동력 ( kW )

$Z_n$  : 커터의 날수 ( teeth/ rev. )

$$\text{또한 } f_{kw} = \frac{4080000 \cdot \eta}{k_s \cdot (X4)^2 \cdot Z_n \cdot N_r} P_{Nominal}$$

#### ③ 축진동 발생 가능성 :

$$\text{IF } Y2 < 0.025 \quad \text{AND } N_r > 2500$$

$$\text{THEN } Y1 = (2.5) \pi X4$$

#### ④ 최대 스피들속도 고정 :

$$\text{IF } N_r > N_{allow} \quad \text{THEN } Y1 = \frac{\pi X4}{1000} N_{allow}$$

여기서  $N_{allow}$  : 허용최대 스피들 속도



⑤ 주변력 ( peripheral force ) 의 제한 :

$$\text{IF } F_p > F_{allow} \quad \text{THEN } Y_2 = \frac{F_{allow}}{0.75 k_s \cdot Z_n \cdot X_4}$$

여기서  $F_p = 0.75 k_s \cdot Z_n \cdot X_4 \cdot Y_2$

$F_{allow}$  : 최대허용 주변력 ( kg )

⑥ 칩발생 불완전요소의 제한 :

$$\text{IF } ( X_3/Y_2 > 20 \text{ OR } X_4/Y_2 < 20 ) \text{ AND } N_r > 2000$$

$$\text{THEN } Y_1 = 2 \pi X_4$$

#### (4) Drilling

① Torque 값의 제한 :

$$\text{IF } M_t > M_{allow} \quad \text{THEN } \bar{Y}_2 = \left( \frac{M_{allow}}{k_d \cdot X_3^2} - 0.63 \right) / 17.0$$

여기서  $M_t = k_d \cdot X_3^2 ( 0.63 + 17.0 Y_2 )$

$k_d$  : NATCO 보정계수 ( 표 참조 )

$M_{allow}$  : Torque 의 최대허용값 ( kg-mm )

표 4-6  $X_1$  값 ( 재질 class 값 ) 에 따른  $k_d$  의 공칭값.

$X_2$ ( 재질 class )	$k_d$ ( 보정계수 )
1, 2, 3, 4, 5, 6, 7	3.44
8, 9, 10, 11, 13	3.20
12, 14, 15, 16, 17	2.65
18, 19, 20, 21	2.32
22, 23, 24, 25, 26	2.12
27, 28, 29, 30	1.80

② Thrust 의 제한

$$\text{IF } T_d > T_{allow} \quad \text{THEN } Y_2 = \left( \frac{T_{allow}}{58.4 \text{ kd } X_3} \right)^{1/0.85}$$

여기서

$$T_d = 58.4 \text{ kd } X_3 (Y_2)^{0.85}$$

$T_{allow}$  : Thrust 의 최대허용값 ( kg )

③ 최대절삭동력의 제한 :

$$\begin{aligned} \text{IF } P_T > P_{allow} \quad \text{THEN} \quad Y_2 &= Y_2 - ( Y_2 - f_{kw} ) / 2 \\ \text{AND} \quad Y_1 &= Y_1 \cdot ( f_{kw} / Y_2 ) \end{aligned}$$

여기서

$$P_T = \frac{1}{\eta} \frac{N_r}{974000} ( \text{kd } X_3^2 ( 0.63 + 17.0 Y_2 ) ) \text{ (kW)}$$

그리고  $\text{kd}$  : 비절삭저항값 (  $\text{kg/mm}^2$  )

$\eta$  : 기계효율 ( 0.7-0.8, 0.75 기준으로 사용 )

$N_r$  : 스피들 회전수 (  $N_r = ( 1000 Y_1 ) / ( \pi X_3 )$  )

$P_{allow}$  : 공작기계의 허용동력 ( kW )

또한

$$f_{kw} = \left( \frac{974000 \cdot \eta \cdot P_{allow}}{\text{kd} \cdot X_3^2 \cdot N_r} - 0.63 \right) / 17.0$$

④ 구멍길이비에 따른 교정 :

$$\text{IF } 2 < L/X_3 \leq 3 \quad \text{THEN} \quad Y_1 = (0.9) Y_1 \quad \text{AND} \quad Y_2 = (0.9) Y_2$$

$$\text{IF } 3 < L/X_3 \leq 4 \quad \text{THEN} \quad Y_1 = (0.8) Y_1 \quad \text{AND} \quad Y_2 = (0.85) Y_2$$

$$\text{IF } 4 < L/X_3 \leq 6 \quad \text{THEN} \quad Y_1 = (0.7) Y_1 \quad \text{AND} \quad Y_2 = (0.8) Y_2$$

$$\text{IF } 6 < L/X_3 \leq 8 \quad \text{THEN} \quad Y_1 = (0.65) Y_1 \quad \text{AND} \quad Y_2 = (0.75) Y_2$$

$$\text{IF } 8 < L/X_3 \quad \text{THEN} \quad Y_1 = (0.6) Y_1 \quad \text{AND} \quad Y_2 = (0.7) Y_2$$

여기서  $L$  : 구멍의 길이 ( mm )

## (5) Boring

### ① Chip 부하 값의 제한 :

$$\text{IF } F_{ch} > F_{cw} \quad \text{THEN } Y2 = F_{cw} / (k_s X3)$$

$$\text{여기서 } F_{ch} = k_s \cdot X3 \cdot Y2$$

$F_{cw}$  : 허용 최대칩부하 ( kg )

### ② Torque 값의 제한 :

$$\text{IF } M_{bt} > M_{bw} \quad \text{THEN } \overline{Y2} = \frac{M_{bw}}{k_s \cdot D_b \cdot X3 \cdot Y2}$$

$$\text{여기서 } M_{bt} = k_s \cdot (D_b/2) \cdot X3 \cdot Y2$$

$D_b$  : 구멍의 직경 ( mm )

$M_{bw}$  : Torque 의 최대허용값 ( kg-mm )

### ③ 축진동 발생 가능성 :

$$\text{IF } ( Y2 < 0.01 \quad \text{OR} \quad X3 < 0.05 ) \quad \text{AND} \quad ( N_r > 2500 )$$

$$\text{THEN } Y1 = (2.5) \pi D_d$$

# 제 5 장 결 론

## 제 1 절 요약

형상정의와 공정정보를 입력으로, 절삭 가공부위를 8개의 가공 feature 로 대별하여 필요한 절삭조건을 산출하는 작업설계 시스템의 소프트웨어를 완성하였으며 작업설계 기능을 지원하는 정보구조를 설계하였고, 이를 통하여 설계에서 공정설계, 작업설계를 거쳐 절삭가공에 이르는 일련의 정보의 변환 및 흐름을 시스템화 하였다.

절삭가공 데이터베이스를 구축하는 데에 있어서는 다중 퍼셉트론 소자들 로 구성된 신경회로망 조직의 DB editor를 개발함으로써 이를 통해 실용 적 절삭계수들로 구성된 절삭성 데이터베이스를 구축하였고, 또한 fuzzy값 처리와 데이터 갱신률에 의해 신뢰성이 증진된 절삭조건 결정프로그램이 메인 모듈에 인터페이스되어 연결되었다.

그리고 이를 수행하는 과정에서 다음과 같은 사항이 검토되었다.

1. 가공제어인자와 가공상태, 그리고 형상정보들과의 상호관계를 통해 작업설계 과정의 기초적인 2단계 매핑관계의 구성이 가능하였다.
2. 다중 퍼셉트론 방법과 CMAC 방법을 혼합하여 사용함으로써 DB editor 신경회로망의 숨은층을 설계하는 데에 실시간 처리 및 정확도의 구비조건을 충족시킬 수 있었고, fuzzy 값 처리 및 modify rule에 의한 절삭조건 결정 전략을 설정함으로써 실제 응용가능한 신뢰도를 갖게 할 수 있었다.
3. 이와 같은 방법들을 사용함으로써 향후 데이터베이스가 새로 보강/갱신 된 수집데이터에 의해 스스로 학습될 수 있는 기능이 부여되고 있으며, 메

인 모듈과는 독립적으로 학습시킬 수가 있어서 원시프로그램과는 소규모의 common 화일만으로 인터페이스시킬 수 있었다.

## 제 2 절 연구 결과의 토의

이 분야는 학교나 연구소에서 연구테마로 도외시 되어 왔고, 실제 직접 수요자인 기업체에서 구입 사용할 수 있는 상용화 된 시스템도 전무한 상태이며, 자체 노력으로, 가공 파라미터를 결정하기 위한 노하우를 일부 정리하기도 하고 작업자의 체험으로 저장되어 있기도 하여 작업설계에 반영하고는 있으나, 종합적으로 이들의 취합, 소프트웨어화 하는 것이 필요하다.

산재된 현장 노하우의 체계적 수집, 효율적인 데이터구조 설계등에 어려움이 있어 그런 데이터베이스를 구축한 기업체가 거의 없다. 금형부품을 포함하여 각주형 가공물의 절삭가공기술을 소프트웨어에 의해 취합하고 이를 뒷받침할 데이터베이스의 틀을 마련한 것에 첫번째 의미가 있고, 최적 OUTPUT의 도출은 시스템의 보완 발전에 따라 이룩될 수 있을 것으로 본다.

선진국의 경우도 CAM시스템의 보조적 데이터화일의 형태나 핸드북데이터를 소프트웨어화 한 형태로 작업설계자에게 도움을 줄 수 있는 시스템은 있으나, 필요한 정보를 내재하고 있으면서 이를 사용하여 작업설계를 자동으로 수행하는 시스템은 상용화되거나 실용화된 사례가 없다.

여기서 개발된 시스템은 절삭가공을 하는 제조업체에 보급함으로써 가공준비 자동화율을 높일 뿐아니라, 숙련된 작업자의 경험을 데이터의 형태로 저장함으로써 기술의 지속적인 축척 및 진보를 이룰 수 있다.

사출금형가공을 위한 CIM시스템의 한 모듈로서 개발되었으나 가공부위를 정의하는 가공 feature를 범용성있게 설정하여, 인터페이스 모듈의 간단한 수정을 거치면서 금형부품 뿐아니라 일반 상자형가공물에도 적용 가능하도록 한 특징이 있어 기능 확장이 용이하다. 그리고 개발된 절삭가공 DB editor는 향후 새로 보강/갱신된 수집데이터에 의해 스스로 학습될 수 있는 기능이 부여되어 있으므로 자체적으로 시스템의 성능을 높이는 틀로 사용될 수 있다.



## References

- [1] Chang, T.C. and Wysk, R.A., 1985, "An Introduction to Automated Process Planning System," Prentice-Hall Inc., N.J.
- [2] Chang, T.C., 1990, "Expert Process Planning for Manufacturing," Addison-Wesley Pub., N.J.
- [3] Ranky, P.G., 1986, "Computer Integrated Manufacturing," Prentice-Hall Inc., N.J.
- [4] Milacic, V.R., 1990, "Intelligent Manufacturing System I & II," Elsevier, Amsterdam.
- [5] Chryssolouris, G. and Chan, S., 1985, "An Integrated Approach to Process Planning and Scheduling," Annals of the CIRP, Vol.34, pp.413-417.
- [6] Siddall, J.N., 1990, "Expert Systems for Engineers," Marcel Decker Inc., N.J. and Basel.
- [7] Wang, M., Zhu, J.Y. and Zhang, Y.Z., 1985, "Fuzzy Pattern Recognition of the Metal Cutting States," Annals of the CIRP, Vol.34, No.1, pp.133-136.
- [8] Balakrishnam, P. and DeVries, M.F., 1985, "Sequential Estimation of Machinability Parameters for Adaptive Optimization of Machinability Data Base Systems," Trans. of ASME, J. of Eng. for Industry, vOL.107, PP.159-166.
- [9] Zust, R. and Taiber, J., 1990, "Knowledge-Based Process Planning System for Prismatic Workpieces in CAD/CAM-Environment," Annals of the CIRP, Vol.39, No.1, pp.493-496.
- [10] Taraman, S.R., 1987, "Machining Data Bank Structure," CASA/SME

Autofact III conf., pp.249-257.

[11] Carlier, J. and Peters, J., 1985, " MOPS - A Machining Centre Operation Planning System," Annal of the CIRP, Vol.34, No.1, pp.409-411.

[12] Cser, I., Juhasz, M. and Szabo, K., 1988, " Computer-Aided Planning of Machining Processes," Advances in CAD/CAM, IFIP, North-Holland Pub., pp.249-260.

[13] 고영은, 1991, "Neural Network를 이용한 최적절삭조건 결정에 관한 연구," 서울대학교 석사학위논문.

[14] 김성근, 1992, "무계중심 복합구간을 이용한 절삭시스템의 정성모델링과 정성추론 기법에 관한 연구," 서울대학교 박사학위논문.

[15] 정성종, 1992, " 적응제어 수치제어 시스템의 개발 - 신경회로망 기법에 의한 절삭계수의 지적인 선정," 대한기계학회논문집, 제16권, 제7호, pp.1223-1233.

[16] Chryssolouris, G. and Guillot, M., 1990, " A Comparison of Statistical and AI Approaches to Selection of Process Parameters in Intelligent Machining," Trans. of the ASME, Eng. for Industry, Vol.112, pp. 122-131.

[17] Rangwala, S.S. and Dornfeld, D.A., 1989, " Learning and Optimization of Machining Operations Using Computing of Neural Networks," IEEE Trans. on Systems, Man, and Cybernetics, Vol.19, No2, pp.299-314.

[18] Pao, Y.H., 1989, " Adaptive Pattern Recognition and Neural Networks," Addison-Wesley, N.Y.

[18] Pao, Y.H., 1988, " A Connectionist Net Approach to Autonomous Machine Learning of Effective Process Control Strategies," Robotics & Computer-Integrated Manufacturing, Vol.4, No3/4, pp.633-642.

- [19] Metcut Research Associates, 1980, " Machining Data HandBook (Vol. 1 & 2)," Machinability Data Center, Cincinnati, Ohio.
- [20] Wick and Droszda, 1998, " Tool and Manufacturing Handbook - Vol. 1 : Machining," SME, 4-th ed.
- [21] 한국야금(주), 1990, " KORLOY catalog," pp.1-252.
- [21] Toshiba Tungaloy Cutting Tool mfg., 1991, " Toshiba Tungaloy Cutting Tools Handbook," pp.1-556.
- [22] 강무진, 박면웅, 1991, " CIM기술에 의한 사출금형공장 자동화 - 설계와 가공의 인터페이스 기술개발," 한국과학기술연구원 특정과제 3차년도 보고서, 과학기술처.

여 백

## 부록 I . Feature별 표준절삭조건 화일 리스트

여 백



[ Type (1) Data in Face Milling Operations for Learning Procedure ]

Columns : X1 X2 X3 X4 Y1 Y2

No. of Data List = 1920

1.000	1.000	1.000	1.000	1.000	56.000	0.100
2.000	1.000	1.000	1.000	2.000	53.000	0.080
3.000	1.000	1.000	1.000	3.000	50.000	0.060
4.000	1.000	1.000	1.000	4.000	47.000	0.040
5.000	1.000	1.000	2.000	1.000	52.000	0.130
6.000	1.000	1.000	2.000	2.000	48.000	0.100
7.000	1.000	1.000	2.000	3.000	45.000	0.080
8.000	1.000	1.000	2.000	4.000	41.000	0.060
9.000	1.000	1.000	3.000	1.000	47.000	0.150
10.000	1.000	1.000	3.000	2.000	43.000	0.120
11.000	1.000	1.000	3.000	3.000	40.000	0.100
12.000	1.000	1.000	3.000	4.000	36.000	0.080
13.000	1.000	1.000	4.000	1.000	42.000	0.180
14.000	1.000	1.000	4.000	2.000	38.000	0.140
15.000	1.000	1.000	4.000	3.000	34.000	0.120
16.000	1.000	1.000	4.000	4.000	30.000	0.100
17.000	1.000	2.000	1.000	1.000	49.000	0.120
18.000	1.000	2.000	1.000	2.000	45.000	0.100
19.000	1.000	2.000	1.000	3.000	47.000	0.080
20.000	1.000	2.000	1.000	4.000	38.000	0.060
21.000	1.000	2.000	2.000	1.000	43.000	0.150
22.000	1.000	2.000	2.000	2.000	40.000	0.120
23.000	1.000	2.000	2.000	3.000	37.000	0.100
24.000	1.000	2.000	2.000	4.000	34.000	0.080
25.000	1.000	2.000	3.000	1.000	38.000	0.170
26.000	1.000	2.000	3.000	2.000	34.000	0.140
27.000	1.000	2.000	3.000	3.000	31.000	0.110
28.000	1.000	2.000	3.000	4.000	28.000	0.080
29.000	1.000	2.000	4.000	1.000	31.000	0.190
30.000	1.000	2.000	4.000	2.000	28.000	0.160
31.000	1.000	2.000	4.000	3.000	25.000	0.140
32.000	1.000	2.000	4.000	4.000	22.000	0.120
33.000	1.000	3.000	1.000	1.000	41.000	0.150
34.000	1.000	3.000	1.000	2.000	37.000	0.120

[ Machining Data in Peripheral End Milling Operations for Learning Procedure ]

Columns : X1 X2 X3 X4 Y1 Y2

No. of Data List = 1920

1.000	1.000	1.000	1.000	1.000	8.000	0.008
2.000	1.000	1.000	1.000	2.000	8.000	0.007
3.000	1.000	1.000	1.000	3.000	7.000	0.009
4.000	1.000	1.000	1.000	4.000	7.000	0.010
5.000	1.000	1.000	2.000	1.000	6.000	0.004
6.000	1.000	1.000	2.000	2.000	6.000	0.006
7.000	1.000	1.000	2.000	3.000	5.000	0.008
8.000	1.000	1.000	2.000	4.000	5.000	0.008
9.000	1.000	1.000	3.000	1.000	4.000	0.004
10.000	1.000	1.000	3.000	2.000	4.000	0.005
11.000	1.000	1.000	3.000	3.000	4.000	0.006
12.000	1.000	1.000	3.000	4.000	4.000	0.007
13.000	1.000	1.000	4.000	1.000	4.000	0.004
14.000	1.000	1.000	4.000	2.000	4.000	0.005
15.000	1.000	1.000	4.000	3.000	4.000	0.006
16.000	1.000	1.000	4.000	4.000	4.000	0.007
17.000	1.000	2.000	1.000	1.000	16.000	0.011
18.000	1.000	2.000	1.000	2.000	16.000	0.011
19.000	1.000	2.000	1.000	3.000	15.000	0.026
20.000	1.000	2.000	1.000	4.000	15.000	0.034
21.000	1.000	2.000	2.000	1.000	12.000	0.008
22.000	1.000	2.000	2.000	2.000	11.000	0.017
23.000	1.000	2.000	2.000	3.000	11.000	0.033
24.000	1.000	2.000	2.000	4.000	11.000	0.048
25.000	1.000	2.000	3.000	1.000	10.000	0.008
26.000	1.000	2.000	3.000	2.000	10.000	0.009
27.000	1.000	2.000	3.000	3.000	10.000	0.032
28.000	1.000	2.000	3.000	4.000	10.000	0.048
29.000	1.000	2.000	4.000	1.000	9.000	0.008
30.000	1.000	2.000	4.000	2.000	9.000	0.009
31.000	1.000	2.000	4.000	3.000	9.000	0.025
32.000	1.000	2.000	4.000	4.000	9.000	0.033
33.000	1.000	3.000	1.000	1.000	19.000	0.012
34.000	1.000	3.000	1.000	2.000	19.000	0.012

[ Type (1) Data in Slot End Milling Operations for Learning Procedure ]

Columns : X1 X2 X3 X4 Y1 Y2

No. of Data List = 1920

1.000	1.000	1.000	1.000	1.000	15.000	0.010
2.000	1.000	1.000	1.000	2.000	15.000	0.010
3.000	1.000	1.000	1.000	3.000	15.000	0.010
4.000	1.000	1.000	1.000	4.000	15.000	0.030
5.000	1.000	1.000	2.000	1.000	14.000	0.010
6.000	1.000	1.000	2.000	2.000	14.000	0.010
7.000	1.000	1.000	2.000	3.000	13.000	0.030
8.000	1.000	1.000	2.000	4.000	13.000	0.050
9.000	1.000	1.000	3.000	1.000	12.000	0.010
10.000	1.000	1.000	3.000	2.000	12.000	0.010
11.000	1.000	1.000	3.000	3.000	11.000	0.020
12.000	1.000	1.000	3.000	4.000	12.000	0.030
13.000	1.000	1.000	4.000	1.000	11.000	0.010
14.000	1.000	1.000	4.000	2.000	10.000	0.010
15.000	1.000	1.000	4.000	3.000	11.000	0.010
16.000	1.000	1.000	4.000	4.000	11.000	0.020
17.000	1.000	2.000	1.000	1.000	15.000	0.000
18.000	1.000	2.000	1.000	2.000	14.000	0.010
19.000	1.000	2.000	1.000	3.000	14.000	0.020
20.000	1.000	2.000	1.000	4.000	14.000	0.040
21.000	1.000	2.000	2.000	1.000	13.000	0.010
22.000	1.000	2.000	2.000	2.000	12.000	0.020
23.000	1.000	2.000	2.000	3.000	12.000	0.040
24.000	1.000	2.000	2.000	4.000	12.000	0.060
25.000	1.000	2.000	3.000	1.000	10.000	0.010
26.000	1.000	2.000	3.000	2.000	6.000	0.010
27.000	1.000	2.000	3.000	3.000	10.000	0.020
28.000	1.000	2.000	3.000	4.000	10.000	0.040
29.000	1.000	2.000	4.000	1.000	8.000	0.010
30.000	1.000	2.000	4.000	2.000	9.000	0.010
31.000	1.000	2.000	4.000	3.000	8.000	0.030
32.000	1.000	2.000	4.000	4.000	7.000	0.040
33.000	1.000	3.000	1.000	1.000	15.000	0.000
34.000	1.000	3.000	1.000	2.000	15.000	0.010

[ Machining Data in Drilling Operations for Learning Procedure ]

Columns : X1 X2 X3 Y1 Y2

No. of Data List = 960

1.000	1.000	1.000	1.000	4.000	0.010
2.000	1.000	1.000	2.000	4.000	0.013
3.000	1.000	1.000	3.000	4.000	0.025
4.000	1.000	1.000	4.000	5.000	0.050
5.000	1.000	1.000	5.000	5.000	0.050
6.000	1.000	1.000	6.000	5.000	0.075
7.000	1.000	1.000	7.000	5.000	0.075
8.000	1.000	1.000	8.000	5.000	0.102
9.000	1.000	2.000	1.000	3.000	0.010
10.000	1.000	2.000	2.000	3.000	0.013
11.000	1.000	2.000	3.000	3.000	0.025
12.000	1.000	2.000	4.000	3.000	0.075
13.000	1.000	2.000	5.000	3.000	0.102
14.000	1.000	2.000	6.000	3.000	0.102
15.000	1.000	2.000	7.000	3.000	0.102
16.000	1.000	2.000	8.000	3.000	0.102
17.000	1.000	3.000	1.000	11.000	0.010
18.000	1.000	3.000	2.000	11.000	0.013
19.000	1.000	3.000	3.000	11.000	0.025
20.000	1.000	3.000	4.000	12.000	0.050
21.000	1.000	3.000	5.000	12.000	0.075
22.000	1.000	3.000	6.000	12.000	0.075
23.000	1.000	3.000	7.000	13.000	0.075
24.000	1.000	3.000	8.000	13.000	0.102
25.000	1.000	4.000	1.000	18.000	0.001
26.000	1.000	4.000	2.000	18.000	0.002
27.000	1.000	4.000	3.000	18.000	0.004
28.000	1.000	4.000	4.000	19.000	0.008
29.000	1.000	4.000	5.000	19.000	0.015
30.000	1.000	4.000	6.000	19.000	0.025
31.000	1.000	4.000	7.000	20.000	0.035
32.000	1.000	4.000	8.000	20.000	0.045
33.000	2.000	1.000	1.000	5.000	0.020
34.000	2.000	1.000	2.000	5.000	0.025

[ Machining Data in Boring Operations for Learning Procedure ]

Columns : X1 X2 X3 Y1 Y2

No. of Data List = 480

1.000	1.000	1.000	1.000	11.000	0.050
2.000	1.000	1.000	2.000	10.000	0.075
3.000	1.000	1.000	3.000	8.000	0.102
4.000	1.000	1.000	4.000	5.000	0.150
5.000	1.000	2.000	1.000	40.000	0.050
6.000	1.000	2.000	2.000	36.000	0.075
7.000	1.000	2.000	3.000	32.000	0.102
8.000	1.000	2.000	4.000	26.000	0.150
9.000	1.000	3.000	1.000	47.000	0.050
10.000	1.000	3.000	2.000	42.000	0.075
11.000	1.000	3.000	3.000	38.000	0.102
12.000	1.000	3.000	4.000	30.000	0.150
13.000	1.000	4.000	1.000	57.000	0.050
14.000	1.000	4.000	2.000	52.000	0.075
15.000	1.000	4.000	3.000	48.000	0.102
16.000	1.000	4.000	4.000	40.000	0.150
17.000	2.000	1.000	1.000	12.000	0.050
18.000	2.000	1.000	2.000	11.000	0.075
19.000	2.000	1.000	3.000	9.000	0.102
20.000	2.000	1.000	4.000	8.000	0.200
21.000	2.000	2.000	1.000	46.000	0.050
22.000	2.000	2.000	2.000	42.000	0.075
23.000	2.000	2.000	3.000	37.000	0.102
24.000	2.000	2.000	4.000	29.000	0.200
25.000	2.000	3.000	1.000	53.000	0.050
26.000	2.000	3.000	2.000	48.000	0.075
27.000	2.000	3.000	3.000	43.000	0.102
28.000	2.000	3.000	4.000	34.000	0.200
29.000	2.000	4.000	1.000	63.000	0.050
30.000	2.000	4.000	2.000	58.000	0.075
31.000	2.000	4.000	3.000	48.000	0.102
32.000	2.000	4.000	4.000	44.000	0.200
33.000	3.000	1.000	1.000	18.000	0.050
34.000	3.000	1.000	2.000	17.000	0.075
35.000	3.000	1.000	3.000	15.000	0.102

[ Machining Data in Reaming Operations for Learning Procedure ]

Columns : X1 X2 X3 Y1 Y2

No. of Data List = 720

1.000	1.000	1.000	1.000	3.000	0.025
2.000	1.000	1.000	2.000	3.000	0.038
3.000	1.000	1.000	3.000	3.000	0.050
4.000	1.000	1.000	4.000	3.000	0.075
5.000	1.000	1.000	5.000	3.000	0.075
6.000	1.000	1.000	6.000	3.000	0.075
7.000	1.000	2.000	1.000	6.000	0.050
8.000	1.000	2.000	2.000	6.000	0.075
9.000	1.000	2.000	3.000	6.000	0.102
10.000	1.000	2.000	4.000	6.000	0.130
11.000	1.000	2.000	5.000	6.000	0.150
12.000	1.000	2.000	6.000	6.000	0.150
13.000	1.000	3.000	1.000	3.000	0.025
14.000	1.000	3.000	2.000	3.000	0.038
15.000	1.000	3.000	3.000	3.000	0.050
16.000	1.000	3.000	4.000	3.000	0.050
17.000	1.000	3.000	5.000	3.000	0.050
18.000	1.000	3.000	6.000	3.000	0.050
19.000	1.000	4.000	1.000	7.000	0.050
20.000	1.000	4.000	2.000	7.000	0.075
21.000	1.000	4.000	3.000	7.000	0.102
22.000	1.000	4.000	4.000	8.000	0.130
23.000	1.000	4.000	5.000	8.000	0.150
24.000	1.000	4.000	6.000	8.000	0.150
25.000	2.000	1.000	1.000	5.000	0.025
26.000	2.000	1.000	2.000	5.000	0.038
27.000	2.000	1.000	3.000	5.000	0.038
28.000	2.000	1.000	4.000	5.000	0.075
29.000	2.000	1.000	5.000	5.000	0.075
30.000	2.000	1.000	6.000	5.000	0.075
31.000	2.000	2.000	1.000	7.000	0.050
32.000	2.000	2.000	2.000	7.000	0.075
33.000	2.000	2.000	3.000	7.000	0.102
34.000	2.000	2.000	4.000	8.000	0.130
35.000	2.000	2.000	5.000	8.000	0.150

[ Machining Data in Counter Boring & Spotfacing Operations  
for Learning Procedure ]

Columns : X1 X2 X3 Y1 Y2

No. of Data List = 480

1.000	1.000	1.000	1.000	4.000	0.050
2.000	1.000	1.000	2.000	4.000	0.050
3.000	1.000	1.000	3.000	4.000	0.050
4.000	1.000	1.000	4.000	4.000	0.075
5.000	1.000	1.000	5.000	4.000	0.075
6.000	1.000	1.000	6.000	4.000	0.102
7.000	1.000	1.000	7.000	4.000	0.102
8.000	1.000	1.000	8.000	5.000	0.130
9.000	1.000	2.000	1.000	9.000	0.050
10.000	1.000	2.000	2.000	10.000	0.050
11.000	1.000	2.000	3.000	10.000	0.050
12.000	1.000	2.000	4.000	10.000	0.075
13.000	1.000	2.000	5.000	10.000	0.075
14.000	1.000	2.000	6.000	10.000	0.102
15.000	1.000	2.000	7.000	10.000	0.102
16.000	1.000	2.000	8.000	11.000	0.130
17.000	2.000	1.000	1.000	5.000	0.050
18.000	2.000	1.000	2.000	5.000	0.050
19.000	2.000	1.000	3.000	5.000	0.050
20.000	2.000	1.000	4.000	5.000	0.075
21.000	2.000	1.000	5.000	5.000	0.075
22.000	2.000	1.000	6.000	5.000	0.102
23.000	2.000	1.000	7.000	5.000	0.102
24.000	2.000	1.000	8.000	6.000	0.130
25.000	2.000	2.000	1.000	10.000	0.050
26.000	2.000	2.000	2.000	10.000	0.050
27.000	2.000	2.000	3.000	11.000	0.050
28.000	2.000	2.000	4.000	11.000	0.075
29.000	2.000	2.000	5.000	11.000	0.075
30.000	2.000	2.000	6.000	11.000	0.102
31.000	2.000	2.000	7.000	11.000	0.102
32.000	2.000	2.000	8.000	12.000	0.130
33.000	3.000	1.000	1.000	10.000	0.075
34.000	3.000	1.000	2.000	10.000	0.075
35.000	3.000	1.000	3.000	11.000	0.075

[ Type (1) Data in Tapping Operation for Learning Procedure ]

Columns : X1 X2 Y1

No. of Data List = 120

1.000	1.000	0.600
1.000	2.000	0.900
1.000	3.000	1.200
1.000	4.000	1.500
2.000	1.000	0.600
2.000	2.000	1.500
2.000	3.000	1.800
2.000	4.000	2.100
3.000	1.000	1.500
3.000	2.000	3.000
3.000	3.000	4.300
3.000	4.000	5.000
4.000	1.000	2.100
4.000	2.000	5.000
4.000	3.000	5.500
4.000	4.000	6.000
5.000	1.000	1.500
5.000	2.000	3.000
5.000	3.000	4.300
5.000	4.000	5.000
6.000	1.000	2.100
6.000	2.000	5.000
6.000	3.000	5.500
6.000	4.000	6.000
7.000	1.000	2.100
7.000	2.000	5.000
7.000	3.000	5.500
7.000	4.000	6.000
8.000	1.000	3.000
8.000	2.000	8.000
8.000	3.000	9.000
8.000	4.000	11.000
9.000	1.000	2.100
9.000	2.000	5.000
9.000	3.000	6.000

## 부록 II DB Editor 의 수행예시와 출력결과 List

여 백



## A. DB Editor 의 convergence 수행예시 ( Face Mill GROUP A 의 경우 )

warning : SAMPLE number modified by 1920

init error : 1448.960200

== 1 th iteration -----

error = 1448.960200 err\_sum = 2348.960200 delta\_w = 0.000000

iter 1 : weight [ 1 0 0 ]

-0.424800 0.424800 : 1453.132921 1443.248241

0.424800 0.950126 : 1443.248241 1434.571331

0.950126 1.274674 : 1434.571331 1429.500414

1.274674 1.475348 : 1429.500414 1426.871836

1.475348 1.599325 : 1426.871836 1425.405093

1.599325 1.675983 : 1425.405093 1424.545122

1.675983 1.723342 : 1424.545122 1424.049777

1.723342 1.752626 : 1424.049777 1423.750124

1.752626 1.770717 : 1423.750124 1423.567542

1.770717 1.781903 : 1423.567542 1423.455623

1.781903 1.788814 : 1423.455623 1423.386852

del\_w: 1.788814 error: 1423.386852 prev\_err: 1448.960200

iter 1 : weight [ 1 0 1 ]

-0.424800 0.424800 : 1424.936775 1423.263496

0.424800 0.950126 : 1423.263496 1423.027661

0.950126 1.274674 : 1423.027661 1423.441016

0.749452 0.950126 : 1422.480960 1423.027661

0.625475 0.749452 : 1422.558302 1422.480960

del\_w: 0.749452 error: 1422.480960 prev\_err: 1423.386852

iter 1 : weight [ 1 0 2 ]

-0.424800 0.424800 : 1423.331638 1422.222589

0.424800 0.950126 : 1422.222589 1422.293683

del\_w: 0.424800 error: 1422.222589 prev\_err: 1422.480960

iter 1 : weight [ 1 0 3 ]

-0.424800 0.424800 : 1419.548179 1425.851669

-0.950126 -0.424800 : 1415.671031 1419.548179

-1.274674 -0.950126 : 1413.853370 1415.671031

-1.475348 -1.274674 : 1412.808855 1413.853370

-1.599325 -1.475348 : 1412.298369 1412.808855

-1.675983 -1.599325 : 1412.037690 1412.298369

-1.723342 -1.675983 : 1411.901115 1412.037690

-1.752626 -1.723342 : 1411.828336 1411.901115

del\_w: -1.752626 error: 1411.828336 prev\_err: 1422.222589



```

iter 1 : weight [ 1 1 0 ]
-0.424800 0.424800 : 1403.268586 1428.023891
-0.950126 -0.424800 : 1397.005564 1403.268586
-1.274674 -0.950126 : 1394.180490 1397.005564
-1.475348 -1.274674 : 1394.801632 1394.180490
-1.274674 -1.150761 : 1394.180490 1394.618753
-1.351356 -1.274674 : 1394.254344 1394.180490
del_w: -1.274674 error: 1394.180490 prev_err: 1411.828336
iter 1 : weight [ 1 1 1 ]
-0.424800 0.424800 : 1402.968048 1395.287994
0.424800 0.950126 : 1395.287994 1418.622411
0.100422 0.424800 : 1395.063199 1395.287994
-0.100253 0.100422 : 1392.740850 1395.063199
-0.224165 -0.100253 : 1393.053913 1392.740850
-0.100253 -0.023570 : 1392.740850 1393.904737
-0.147538 -0.100253 : 1392.161054 1392.740850
-0.176831 -0.147538 : 1392.192097 1392.161054
del_w: -0.147538 error: 1392.161054 prev_err: 1394.180490
iter 1 : weight [ 1 1 2 ]
-0.424800 0.424800 : 1410.054403 1401.557676
0.424800 0.950126 : 1401.557676 1419.952814
0.100422 0.424800 : 1392.905806 1401.557676
-0.100253 0.100422 : 1396.059017 1392.905806
0.100422 0.224230 : 1392.905806 1395.870894
0.023700 0.100422 : 1392.040650 1392.905806
-0.023595 0.023700 : 1392.600187 1392.040650
0.023700 0.053047 : 1392.040650 1392.242458
0.005682 0.023700 : 1392.097766 1392.040650
del_w: 0.023700 error: 1392.040650 prev_err: 1392.161054
iter 1 : weight [ 1 1 3 ]
-0.424800 0.424800 : 1397.725253 1412.908432
-0.950126 -0.424800 : 1395.634161 1397.725253
-1.274674 -0.950126 : 1391.289343 1395.634161
-1.475348 -1.274674 : 1389.462115 1391.289343
-1.599325 -1.475348 : 1388.413798 1389.462115
-1.675983 -1.599325 : 1387.860385 1388.413798
-1.723342 -1.675983 : 1387.523093 1387.860385
-1.752626 -1.723342 : 1387.311845 1387.523093
-1.770717 -1.752626 : 1387.179929 1387.311845
-1.781903 -1.770717 : 1387.097766 1387.179929
del_w: -1.781903 error: 1387.097766 prev_err: 1392.040650

```

```

iter 1 : weight [ 1 2 0 ]
-0.424800 0.424800 : 1427.477883 1338.999215
0.424800 0.950126 : 1338.999215 1276.391269
0.950126 1.274674 : 1276.391269 1257.409256
1.274674 1.475348 : 1257.409256 1249.355860
1.475348 1.599325 : 1249.355860 1245.359848
1.599325 1.675983 : 1245.359848 1243.287683
1.675983 1.723342 : 1243.287683 1242.138880
1.723342 1.752626 : 1242.138880 1241.456918
1.752626 1.770717 : 1241.456918 1241.051716
1.770717 1.781903 : 1241.051716 1240.802567
1.781903 1.788814 : 1240.802567 1240.649176
1.788814 1.793087 : 1240.649176 1240.554534
del_w: 1.793087 error: 1240.554534 prev_err: 1387.097766
iter 1 : weight [ 1 2 1 ]
-0.424800 0.424800 : 1261.327371 1232.145476
0.424800 0.950126 : 1232.145476 1243.509643
0.100422 0.424800 : 1238.663397 1232.145476
0.424800 0.625539 : 1232.145476 1234.920506
0.301017 0.424800 : 1233.438551 1232.145476
0.424800 0.501572 : 1232.145476 1232.484433
0.377629 0.424800 : 1232.188928 1232.145476
del_w: 0.424800 error: 1232.145476 prev_err: 1240.554534
iter 1 : weight [ 1 2 2 ]
-0.424800 0.424800 : 1235.744644 1231.275602
0.424800 0.950126 : 1231.275602 1235.188609
0.100422 0.424800 : 1231.708138 1231.275602
0.424800 0.625539 : 1231.275602 1232.512378
0.301017 0.424800 : 1231.280641 1231.275602
del_w: 0.424800 error: 1231.275602 prev_err: 1232.145476
iter 1 : weight [ 1 2 3 ]
-0.424800 0.424800 : 1236.684055 1227.634275
0.424800 0.950126 : 1227.634275 1222.109075
0.950126 1.274674 : 1222.109075 1218.311689
1.274674 1.475348 : 1218.311689 1216.591058
1.475348 1.599325 : 1216.591058 1216.132731
1.599325 1.675983 : 1216.132731 1215.745499
1.675983 1.723342 : 1215.745499 1215.468073
1.723342 1.752626 : 1215.468073 1215.324748
1.752626 1.770717 : 1215.324748 1215.251304
del_w: 1.770717 error: 1215.251304 prev_err: 1231.275602

```

```

-0.100253 -0.023570 : 720.429079 720.187759
-0.023570 0.023764 : 720.187759 720.249146
del_w: -0.000300 error: 720.182089 prev_err: 720.181932
iter 1 : weight [ 2 0 2 ]
-0.424800 0.424800 : 722.202987 725.084965
-0.950126 -0.424800 : 726.113297 722.202987
-0.424800 -0.100422 : 722.202987 720.233608
-0.100422 0.100253 : 720.233608 720.600079
-0.224230 -0.100422 : 720.829413 720.233608
-0.100422 -0.023700 : 720.233608 720.133762
del_w: -0.023700 error: 720.133762 prev_err: 720.182089
iter 1 : weight [ 2 0 3 ]
-0.424800 0.424800 : 745.636279 714.358785
0.424800 0.950126 : 714.358785 703.799174
0.950126 1.274674 : 703.799174 698.655981
1.274674 1.475348 : 698.655981 697.021732
1.475348 1.599325 : 697.021732 696.372053
1.599325 1.675983 : 696.372053 695.995434
1.675983 1.723342 : 695.995434 695.727552
1.723342 1.752626 : 695.727552 695.552792
1.752626 1.770717 : 695.552792 695.459910
del_w: 1.770717 error: 695.459910 prev_err: 720.133762
iter 1 : weight [ 2 0 4 ]
-0.424800 0.424800 : 737.224509 728.363874
0.424800 0.950126 : 728.363874 732.915147
0.100422 0.424800 : 701.467076 728.363874
-0.100253 0.100422 : 691.976088 701.467076
-0.224165 -0.100253 : 696.608340 691.976088
-0.100253 -0.023570 : 691.976088 695.252178
-0.147538 -0.100253 : 693.217409 691.976088
-0.100253 -0.070926 : 691.976088 693.042899
-0.118272 -0.100253 : 692.032519 691.976088
del_w: -0.100253 error: 691.976088 prev_err: 695.459910
iter 1 : weight [ 2 0 5 ]
-0.424800 0.424800 : 788.812763 756.926443
0.424800 0.950126 : 756.926443 767.735138
0.100422 0.424800 : 689.832879 756.926443
-0.100253 0.100422 : 700.829215 689.832879
0.100422 0.224230 : 689.832879 695.361493
0.023700 0.100422 : 691.574564 689.832879
0.100422 0.147627 : 689.832879 689.818453

```

```

del_w: 0.147627 error: 689.818453 prev_err: 691.976088
iter 1 : weight [ 2 0 6 ]
-0.424800 0.424800 : 688.814785 786.559682
-0.950126 -0.424800 : 689.044603 688.814785
-0.424800 -0.100422 : 688.814785 689.956308
-0.625539 -0.424800 : 688.104381 688.814785
-0.749452 -0.625539 : 688.289936 688.104381
-0.625539 -0.548817 : 688.104381 688.096981
del_w: -0.548817 error: 688.096981 prev_err: 689.818453
iter 1 : weight [ 2 1 0 ]
-0.424800 0.424800 : 696.872294 691.737686
0.424800 0.950126 : 691.737686 709.197459
0.100422 0.424800 : 689.103014 691.737686
-0.100253 0.100422 : 686.393828 689.103014
-0.224165 -0.100253 : 690.466381 686.393828
-0.100253 -0.023570 : 686.393828 687.390849
-0.147538 -0.100253 : 687.050169 686.393828
-0.100253 -0.070926 : 686.393828 686.503137
-0.118272 -0.100253 : 686.533034 686.393828
-0.100253 -0.089012 : 686.393828 686.390118
del_w: -0.089012 error: 686.390118 prev_err: 688.096981
iter 1 : weight [ 2 1 1 ]
-0.424800 0.424800 : 689.816082 693.011848
-0.950126 -0.424800 : 688.868086 689.816082
-1.274674 -0.950126 : 690.628654 688.868086
-0.950126 -0.749452 : 688.868086 689.516270
-1.074039 -0.950126 : 688.863641 688.868086
del_w: -0.000300 error: 686.389597 prev_err: 686.390118
iter 1 : weight [ 2 1 2 ]
-0.424800 0.424800 : 691.907814 688.048747
0.424800 0.950126 : 688.048747 686.416902
0.950126 1.274674 : 686.416902 685.539110
1.274674 1.475348 : 685.539110 686.604621
1.150761 1.274674 : 685.684363 685.539110
1.274674 1.351356 : 685.539110 685.785363
1.227388 1.274674 : 685.527123 685.539110
del_w: 1.227388 error: 685.527123 prev_err: 686.389597
iter 1 : weight [ 2 1 3 ]
-0.424800 0.424800 : 684.086825 687.695258
-0.950126 -0.424800 : 683.020926 684.086825
-1.274674 -0.950126 : 682.581425 683.020926

```

```

-0.005529 0.005682 : 651.378621 650.535331
0.005682 0.012534 : 650.535331 650.777700
0.001372 0.005682 : 650.638612 650.535331
0.005682 0.008270 : 650.535331 650.585093
del_w: 0.005682 error: 650.535331 prev_err: 650.747387
iter 1 : weight [ 2 4 0 ]
-0.424800 0.424800 : 698.523044 709.862366
-0.950126 -0.424800 : 713.602609 698.523044
-0.424800 -0.100422 : 698.523044 648.283320
-0.100422 0.100253 : 648.283320 666.459458
-0.224230 -0.100422 : 656.405236 648.283320
-0.100422 -0.023700 : 648.283320 648.378875
del_w: -0.100422 error: 648.283320 prev_err: 650.535331
iter 1 : weight [ 2 4 1 ]
-0.424800 0.424800 : 648.788071 654.746188
-0.950126 -0.424800 : 662.648317 648.788071
-0.424800 -0.100422 : 648.788071 647.523392
-0.100422 0.100253 : 647.523392 649.337526
-0.224230 -0.100422 : 647.147533 647.523392
-0.300888 -0.224230 : 647.346319 647.147533
-0.224230 -0.177000 : 647.147533 647.165555
del_w: -0.224230 error: 647.147533 prev_err: 648.283320
iter 1 : weight [ 2 4 2 ]
-0.424800 0.424800 : 647.471005 646.961114
0.424800 0.950126 : 646.961114 647.115087
0.100422 0.424800 : 647.087227 646.961114
0.424800 0.625539 : 646.961114 646.962744
del_w: 0.424800 error: 646.961114 prev_err: 647.147533
iter 1 : weight [ 2 4 3 ]
-0.424800 0.424800 : 648.608900 647.487379
0.424800 0.950126 : 647.487379 650.649606
0.100422 0.424800 : 646.868078 647.487379
-0.100253 0.100422 : 647.175407 646.868078
0.100422 0.224230 : 646.868078 646.948429
del_w: 0.100422 error: 646.868078 prev_err: 646.961114
iter 1 : weight [ 2 4 4 ]
-0.424800 0.424800 : 777.991869 654.026521
0.424800 0.950126 : 654.026521 654.553896
0.100422 0.424800 : 651.340655 654.026521
-0.100253 0.100422 : 661.744120 651.340655
0.100422 0.224230 : 651.340655 653.332319

```

```

0.023700 0.100422 : 647.028432 651.340655
-0.023595 0.023700 : 648.090109 647.028432
0.023700 0.053047 : 647.028432 648.782893
0.005682 0.023700 : 646.818058 647.028432
-0.005529 0.005682 : 647.026906 646.818058
0.005682 0.012534 : 646.818058 646.810408
del_w: 0.012534 error: 646.810408 prev_err: 646.868078
iter 1 : weight [ 2 4 5 ]
-0.424800 0.424800 : 776.045991 662.400004
0.424800 0.950126 : 662.400004 663.021800
0.100422 0.424800 : 651.898469 662.400004
-0.100253 0.100422 : 662.498003 651.898469
0.100422 0.224230 : 651.898469 656.250845
0.023700 0.100422 : 647.535511 651.898469
-0.023595 0.023700 : 647.859272 647.535511
0.023700 0.053047 : 647.535511 649.676021
0.005682 0.023700 : 646.856160 647.535511
-0.005529 0.005682 : 646.848925 646.856160
del_w: -0.000300 error: 646.809709 prev_err: 646.810408
iter 1 : weight [ 2 4 6 ]
-0.424800 0.424800 : 676.662111 718.180502
-0.950126 -0.424800 : 723.818346 676.662111
-0.424800 -0.100422 : 676.662111 652.658712
-0.100422 0.100253 : 652.658712 654.971542
-0.224230 -0.100422 : 656.033874 652.658712
-0.100422 -0.023700 : 652.658712 648.011551
-0.023700 0.023595 : 648.011551 646.726398
0.023595 0.052903 : 646.726398 648.193683
0.005563 0.023595 : 646.707655 646.726398
del_w: 0.005563 error: 646.707655 prev_err: 646.809709
iter 1 : weight [ 2 5 0 ]
-0.424800 0.424800 : 650.734943 655.663136
-0.950126 -0.424800 : 670.193973 650.734943
-0.424800 -0.100422 : 650.734943 643.873957
-0.100422 0.100253 : 643.873957 649.226244
-0.224230 -0.100422 : 641.688948 643.873957
-0.300888 -0.224230 : 643.310691 641.688948
-0.224230 -0.177000 : 641.688948 642.122708
-0.253562 -0.224230 : 642.012804 641.688948
-0.224230 -0.206247 : 641.688948 641.801358
-0.235488 -0.224230 : 641.728589 641.688948

```



```

-0.424800 0.424800 : 1595.827721 847.564102
0.424800 0.950126 : 847.564102 848.414213
0.100422 0.424800 : 837.004148 847.564102
-0.100253 0.100422 : 1505.755411 837.004148
0.100422 0.224230 : 837.004148 846.657134
0.023700 0.100422 : 729.054543 837.004148
-0.023595 0.023700 : 777.893798 729.054543
0.023700 0.053047 : 729.054543 805.201098
0.005682 0.023700 : 651.735972 729.054543
-0.005529 0.005682 : 640.022763 651.735972
-0.012411 -0.005529 : 674.364371 640.022763
-0.005529 -0.001229 : 640.022763 635.390345
-0.001229 0.001400 : 635.390345 638.817657
-0.002882 -0.001229 : 635.646345 635.390345
-0.001229 -0.000236 : 635.390345 636.193283
-0.001871 -0.001229 : 635.248379 635.390345
-0.002251 -0.001871 : 635.311082 635.248379
del_w: -0.001871 error: 635.248379 prev_err: 636.493019
iter 1 : weight [ 3 0 3 ]
-0.424800 0.424800 : 1149.117550 762.813097
0.424800 0.950126 : 762.813097 788.794167
0.100422 0.424800 : 655.771879 762.813097
-0.100253 0.100422 : 706.420949 655.771879
0.100422 0.224230 : 655.771879 713.315140
0.023700 0.100422 : 633.886020 655.771879
-0.023595 0.023700 : 642.710256 633.886020
0.023700 0.053047 : 633.886020 638.495913
0.005682 0.023700 : 634.451199 633.886020
0.023700 0.034954 : 633.886020 634.874370
0.016864 0.023700 : 633.783718 633.886020
0.012565 0.016864 : 633.904016 633.783718
0.016864 0.019446 : 633.783718 633.785175
del_w: 0.016864 error: 633.783718 prev_err: 635.248379
iter 1 : weight [ 3 0 4 ]
-0.424800 0.424800 : 1616.985659 811.269302
0.424800 0.950126 : 811.269302 815.902207
0.100422 0.424800 : 771.048152 811.269302
-0.100253 0.100422 : 1201.163702 771.048152
0.100422 0.224230 : 771.048152 803.940796
0.023700 0.100422 : 668.717435 771.048152
-0.023595 0.023700 : 698.451153 668.717435

```

```

0.023700 0.053047 : 668.717435 728.423381
0.005682 0.023700 : 634.068507 668.717435
-0.005529 0.005682 : 639.273569 634.068507
0.005682 0.012534 : 634.068507 642.429535
0.001372 0.005682 : 633.242997 634.068507
-0.001246 0.001372 : 634.546719 633.242997
0.001372 0.003036 : 633.242997 633.038033
0.003036 0.004036 : 633.038033 633.212943
0.002389 0.003036 : 633.064000 633.038033
del_w: 0.003036 error: 633.038033 prev_err: 633.783718
iter 1 : weight [ 3 0 5 ]
-0.424800 0.424800 : 1564.844439 860.743636
0.424800 0.950126 : 860.743636 865.038688
0.100422 0.424800 : 833.219887 860.743636
-0.100253 0.100422 : 1387.490869 833.219887
0.100422 0.224230 : 833.219887 853.846894
0.023700 0.100422 : 703.315326 833.219887
-0.023595 0.023700 : 732.680281 703.315326
0.023700 0.053047 : 703.315326 776.787089
0.005682 0.023700 : 640.200598 703.315326
-0.005529 0.005682 : 637.907936 640.200598
-0.012411 -0.005529 : 660.884160 637.907936
-0.005529 -0.001229 : 637.907936 633.100963
-0.001229 0.001400 : 633.100963 633.771540
-0.002882 -0.001229 : 634.090706 633.100963
-0.001229 -0.000236 : 633.100963 633.000158
-0.000236 0.000395 : 633.000158 633.166399
-0.000609 -0.000236 : 633.002758 633.000158
del_w: -0.000236 error: 633.000158 prev_err: 633.038033
iter 1 : weight [ 3 0 6 ]
-0.424800 0.424800 : 864.173287 1106.355957
-0.950126 -0.424800 : 933.804315 864.173287
-0.424800 -0.100422 : 864.173287 684.746011
-0.100422 0.100253 : 684.746011 697.024649
-0.224230 -0.100422 : 777.588079 684.746011
-0.100422 -0.023700 : 684.746011 637.307002
-0.023700 0.023595 : 637.307002 636.631372
0.023595 0.052903 : 636.631372 652.573542
0.005563 0.023595 : 633.226849 636.631372
-0.005633 0.005563 : 633.281123 633.226849
del_w: -0.000300 error: 633.004518 prev_err: 633.000158

```

## B. 절삭계수 데이터베이스의 출력결과 예시

[ FEED RATE DATA\_FILE : FACE MILLING, DATA\_GROUP : A ]

current error : 93.854509  
SUCCESS in LEARNING WEIGHT at 3 iteration

```

num_layer: 4
num_node: 3 7 7 1
average: 15.500000 2.500000 2.500000 0.207229
scale: 18.850000 1.950000 1.950000 0.217398
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: 2.625200
weight[1][0][1]: 9.616808
weight[1][0][2]: 4.784286
weight[1][1][0]: 8.689397
weight[1][1][1]: 7.516827
weight[1][1][2]: 17.860443
weight[1][2][0]: 3.168229
weight[1][2][1]: -5.508467
weight[1][2][2]: -2.901814
weight[1][3][0]: -0.455763
weight[1][3][1]: -11.155880
weight[1][3][2]: -8.514910
weight[1][4][0]: -4.271378
weight[1][4][1]: -9.344964
weight[1][4][2]: -9.837268
weight[1][5][0]: 7.403172
weight[1][5][1]: 3.009906
weight[1][5][2]: 9.966645
weight[1][6][0]: -9.517643
weight[1][6][1]: -6.401173
weight[1][6][2]: 2.652363
weight[2][0][0]: 5.328442
weight[2][0][1]: 7.276323
weight[2][0][2]: -0.490472
weight[2][0][3]: 1.023090
weight[2][0][4]: -4.719570
weight[2][0][5]: -9.430909
weight[2][0][6]: -4.712750
weight[2][1][0]: 9.563049
weight[2][1][1]: -4.175496
weight[2][1][2]: -6.783577
weight[2][1][3]: -1.482080
weight[2][1][4]: -7.005798
weight[2][1][5]: 0.190333
weight[2][1][6]: -7.600915
weight[2][2][0]: 4.855213
weight[2][2][1]: 7.959468
weight[2][2][2]: -12.929333
weight[2][2][3]: 2.555312
weight[2][2][4]: 2.037910
weight[2][2][5]: -10.997953
weight[2][2][6]: -8.491683
weight[2][3][0]: -0.020555
weight[2][3][1]: -9.781319
weight[2][3][2]: 2.155867
weight[2][3][3]: -6.423798
weight[2][3][4]: 8.726408
weight[2][3][5]: 4.353609
weight[2][3][6]: -9.433974
weight[2][4][0]: -2.162023
weight[2][4][1]: -8.586581
weight[2][4][2]: 8.742493
weight[2][4][3]: -5.679035
weight[2][4][4]: -2.445407
weight[2][4][5]: 0.846409
weight[2][4][6]: 7.150319
weight[2][5][0]: -10.733198
weight[2][5][1]: -7.487904
weight[2][5][2]: 6.896206
weight[2][5][3]: -7.124346
weight[2][5][4]: 9.598070
weight[2][5][5]: -4.494306
weight[2][5][6]: -9.934380
weight[2][6][0]: 7.355150
weight[2][6][1]: -2.870458
weight[2][6][2]: 8.857106
weight[2][6][3]: -7.670138
weight[2][6][4]: 4.763885
weight[2][6][5]: -3.282846
weight[2][6][6]: 4.643467
weight[3][0][0]: 8.359359
weight[3][0][1]: -5.070630
weight[3][0][2]: -0.859826
weight[3][0][3]: 9.072904
weight[3][0][4]: -9.293105
weight[3][0][5]: -0.257024
weight[3][0][6]: 0.835675
network_fuzzy_number: 0.460000

```

[ FEED RATE DATA\_FILE : FACE MILLING, DATA\_GROUP : B ]

current error : 83.718222  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.500000 2.500000 2.500000 0.207229  
scale: 18.850000 1.950000 1.950000 0.217398  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight{1}{0}{0}: 2.624800  
weight{1}{0}{1}: 9.617448  
weight{1}{0}{2}: 4.785086  
weight{1}{1}{0}: 8.689517  
weight{1}{1}{1}: 7.516167  
weight{1}{1}{2}: 17.861343  
weight{1}{2}{0}: 3.167529  
weight{1}{2}{1}: -5.508507  
weight{1}{2}{2}: -2.902294  
weight{1}{3}{0}: -0.456683  
weight{1}{3}{1}: -11.155720  
weight{1}{3}{2}: -8.514490  
weight{1}{4}{0}: -4.270798  
weight{1}{4}{1}: -9.344124  
weight{1}{4}{2}: -9.837068  
weight{1}{5}{0}: 7.402412  
weight{1}{5}{1}: 3.009326  
weight{1}{5}{2}: 9.966905  
weight{1}{6}{0}: -9.517703  
weight{1}{6}{1}: -6.401793  
weight{1}{6}{2}: 2.652183  
weight{2}{0}{0}: 5.329242  
weight{2}{0}{1}: 7.277023  
weight{2}{0}{2}: -0.491192  
weight{2}{0}{3}: 1.022270  
weight{2}{0}{4}: -4.719530  
weight{2}{0}{5}: -9.430489  
weight{2}{0}{6}: -4.712170  
weight{2}{1}{0}: 9.562369  
weight{2}{1}{1}: -4.174876  
weight{2}{1}{2}: -6.783557  
weight{2}{1}{3}: -1.481180  
weight{2}{1}{4}: -7.004938  
weight{2}{1}{5}: 0.190013  
weight{2}{1}{6}: -7.601715  
weight{2}{2}{0}: 4.855793  
weight{2}{2}{1}: 7.960368  
weight{2}{2}{2}: -12.929113  
weight{2}{2}{3}: 2.556152  
weight{2}{2}{4}: 2.038690  
weight{2}{2}{5}: -10.997193  
weight{2}{2}{6}: -8.491363  
weight{2}{3}{0}: -0.020275  
weight{2}{3}{1}: -9.780479  
weight{2}{3}{2}: 2.156127  
weight{2}{3}{3}: -6.423478  
weight{2}{3}{4}: 8.726688  
weight{2}{3}{5}: 4.353389  
weight{2}{3}{6}: -9.433954  
weight{2}{4}{0}: -2.162483  
weight{2}{4}{1}: -8.587581  
weight{2}{4}{2}: 8.743393  
weight{2}{4}{3}: -5.679795  
weight{2}{4}{4}: -2.446247  
weight{2}{4}{5}: 0.846729  
weight{2}{4}{6}: 7.150259  
weight{2}{5}{0}: -10.733358  
weight{2}{5}{1}: -7.487424  
weight{2}{5}{2}: 6.896586  
weight{2}{5}{3}: -7.123566  
weight{2}{5}{4}: 9.598730  
weight{2}{5}{5}: -4.493986  
weight{2}{5}{6}: -9.934560  
weight{2}{6}{0}: 7.355950  
weight{2}{6}{1}: -2.869898  
weight{2}{6}{2}: 8.857406  
weight{2}{6}{3}: -7.669558  
weight{2}{6}{4}: 4.764685  
weight{2}{6}{5}: -3.283186  
weight{2}{6}{6}: 4.643527  
weight{3}{0}{0}: 8.358939  
weight{3}{0}{1}: -5.069930  
weight{3}{0}{2}: -0.860386  
weight{3}{0}{3}: 9.072564  
weight{3}{0}{4}: -9.293365  
weight{3}{0}{5}: -0.257304  
weight{3}{0}{6}: 0.836035  
network\_fuzzy\_number: 0.600000



[ VELOCITY DATA\_FILE : FACE MILLING, DATA\_GROUP : A ]

current error : 116.352286  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.500000 2.500000 2.500000 153.920837  
scale: 18.850000 1.950000 1.950000 161.097092  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: 2.378416  
weight[1][0][1]: -13.070593  
weight[1][0][2]: 5.315570  
weight[1][1][0]: 0.360977  
weight[1][1][1]: -1.526786  
weight[1][1][2]: 5.233820  
weight[1][2][0]: 4.038644  
weight[1][2][1]: -9.630789  
weight[1][2][2]: 1.514060  
weight[1][3][0]: 8.847868  
weight[1][3][1]: -4.877703  
weight[1][3][2]: -2.455104  
weight[1][4][0]: -15.296268  
weight[1][4][1]: -0.818386  
weight[1][4][2]: 3.878979  
weight[1][5][0]: -3.181122  
weight[1][5][1]: 2.546511  
weight[1][5][2]: -5.706377  
weight[1][6][0]: 36.445683  
weight[1][6][1]: 11.100704  
weight[1][6][2]: 2.590263  
weight[2][0][0]: 8.933540  
weight[2][0][1]: -6.336108  
weight[2][0][2]: 5.215755  
weight[2][0][3]: 5.484618  
weight[2][0][4]: 4.340557  
weight[2][0][5]: -1.892041  
weight[2][0][6]: 3.829532  
weight[2][1][0]: 1.935779  
weight[2][1][1]: -0.303563  
weight[2][1][2]: 4.218148  
weight[2][1][3]: 0.215961  
weight[2][1][4]: -8.619526  
weight[2][1][5]: -3.278571  
weight[2][1][6]: -11.586485  
weight[2][2][0]: 9.330683  
weight[2][2][1]: 7.744437  
weight[2][2][2]: 15.361176  
weight[2][2][3]: 6.207135  
weight[2][2][4]: 3.838667  
weight[2][2][5]: 6.776310  
weight[2][2][6]: -6.547143  
weight[2][3][0]: 0.026001  
weight[2][3][1]: -8.022769  
weight[2][3][2]: -7.099641  
weight[2][3][3]: -3.206596  
weight[2][3][4]: 10.037685  
weight[2][3][5]: -3.454368  
weight[2][3][6]: 2.956632  
weight[2][4][0]: -1.314086  
weight[2][4][1]: 4.136955  
weight[2][4][2]: -10.303006  
weight[2][4][3]: -7.046172  
weight[2][4][4]: 1.406902  
weight[2][4][5]: 5.134300  
weight[2][4][6]: -2.154473  
weight[2][5][0]: -7.195876  
weight[2][5][1]: 2.817897  
weight[2][5][2]: 2.149819  
weight[2][5][3]: 0.259440  
weight[2][5][4]: -1.646152  
weight[2][5][5]: -2.138044  
weight[2][5][6]: 3.897979  
weight[2][6][0]: 4.801625  
weight[2][6][1]: -3.185573  
weight[2][6][2]: -12.590962  
weight[2][6][3]: -5.598734  
weight[2][6][4]: -5.708015  
weight[2][6][5]: -5.608322  
weight[2][6][6]: -0.530561  
weight[3][0][0]: 8.971735  
weight[3][0][1]: 3.957507  
weight[3][0][2]: -8.870873  
weight[3][0][3]: 0.196260  
weight[3][0][4]: -4.809511  
weight[3][0][5]: 5.069622  
weight[3][0][6]: 6.727710  
network\_fuzzy\_number: 0.460000



[ VELOCITY DATA\_FILE : FACE MILLING, DATA\_GROUP : B ]

current error : 103.786239  
 SUCCESS in LEARNING WEIGHT at 3 iteration

```

num_layer: 4
num_node: 3 7 7 1
average: 15.500000 2.500000 2.500000 153.920837
scale: 18.850000 1.950000 1.950000 161.097092
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: 2.378016
weight[1][0][1]: -13.069953
weight[1][0][2]: 5.316370
weight[1][1][0]: 0.361097
weight[1][1][1]: -1.527446
weight[1][1][2]: 5.234720
weight[1][2][0]: 4.037944
weight[1][2][1]: -9.630829
weight[1][2][2]: 1.513580
weight[1][3][0]: 8.846948
weight[1][3][1]: -4.877543
weight[1][3][2]: -2.454684
weight[1][4][0]: -15.295688
weight[1][4][1]: -0.817546
weight[1][4][2]: 3.879179
weight[1][5][0]: -3.181882
weight[1][5][1]: 2.545931
weight[1][5][2]: -5.706117
weight[1][6][0]: 36.445623
weight[1][6][1]: 11.100084
weight[1][6][2]: 2.590083
weight[2][0][0]: 8.934340
weight[2][0][1]: -6.335408
weight[2][0][2]: 5.215035
weight[2][0][3]: 5.483798
weight[2][0][4]: 4.340597
weight[2][0][5]: -1.891621
weight[2][0][6]: 3.830112
weight[2][1][0]: 1.935099
weight[2][1][1]: -0.302943
weight[2][1][2]: 4.218168
weight[2][1][3]: 0.216861
weight[2][1][4]: -8.618666
weight[2][1][5]: -3.278891
weight[2][1][6]: -11.587285
weight[2][2][0]: 9.331263
weight[2][2][1]: 7.745337
weight[2][2][2]: 15.361396
weight[2][2][3]: 6.207975
weight[2][2][4]: 3.839447
weight[2][2][5]: 6.777070
weight[2][2][6]: -6.546823
weight[2][3][0]: 0.026281
weight[2][3][1]: -8.021929
weight[2][3][2]: -7.099381
weight[2][3][3]: -3.206276
weight[2][3][4]: 10.037965
weight[2][3][5]: -3.454588
weight[2][3][6]: 2.956652
weight[2][4][0]: -1.314546
weight[2][4][1]: 4.135955
weight[2][4][2]: -10.302106
weight[2][4][3]: -7.046932
weight[2][4][4]: 1.406062
weight[2][4][5]: 5.134620
weight[2][4][6]: -2.154533
weight[2][5][0]: -7.196036
weight[2][5][1]: 2.818377
weight[2][5][2]: 2.150199
weight[2][5][3]: 0.260220
weight[2][5][4]: -1.645492
weight[2][5][5]: -2.137724
weight[2][5][6]: 3.897799
weight[2][6][0]: 4.802425
weight[2][6][1]: -3.185013
weight[2][6][2]: -12.590662
weight[2][6][3]: -5.598154
weight[2][6][4]: -5.707215
weight[2][6][5]: -5.608662
weight[2][6][6]: -0.530501
weight[3][0][0]: 8.971315
weight[3][0][1]: 3.958207
weight[3][0][2]: -8.871433
weight[3][0][3]: 0.195920
weight[3][0][4]: -4.809771
weight[3][0][5]: 5.069342
weight[3][0][6]: 6.728070
network_fuzzy_number: 0.600000

```

[ FEED RATE DATA\_FILE : END MILLING, DATA\_GROUP : A ]

current error : 111.181504  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.689655 2.500000 2.500000 0.062278  
scale: 18.603449 1.950000 1.950000 0.075761  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: -2.845333  
weight[1][0][1]: -7.653771  
weight[1][0][2]: -1.718533  
weight[1][1][0]: -6.098242  
weight[1][1][1]: 2.193134  
weight[1][1][2]: 12.498396  
weight[1][2][0]: 7.666320  
weight[1][2][1]: -10.609221  
weight[1][2][2]: 3.808480  
weight[1][3][0]: 2.917535  
weight[1][3][1]: -17.976295  
weight[1][3][2]: 4.481180  
weight[1][4][0]: -1.152853  
weight[1][4][1]: -7.315074  
weight[1][4][2]: -1.281712  
weight[1][5][0]: -5.283306  
weight[1][5][1]: -2.956119  
weight[1][5][2]: -6.769426  
weight[1][6][0]: 10.807340  
weight[1][6][1]: 5.486214  
weight[1][6][2]: 6.211746  
weight[2][0][0]: 3.993208  
weight[2][0][1]: 4.660013  
weight[2][0][2]: 4.718709  
weight[2][0][3]: 6.914861  
weight[2][0][4]: -8.377044  
weight[2][0][5]: 11.060924  
weight[2][0][6]: -3.179402  
weight[2][1][0]: -6.985947  
weight[2][1][1]: -6.879461  
weight[2][1][2]: -4.986746  
weight[2][1][3]: -7.226872  
weight[2][1][4]: -7.013579  
weight[2][1][5]: 4.127073  
weight[2][1][6]: -5.152562  
weight[2][2][0]: -4.226021  
weight[2][2][1]: 2.178013  
weight[2][2][2]: -3.617895  
weight[2][2][3]: -1.087657  
weight[2][2][4]: -1.534396  
weight[2][2][5]: 2.840471  
weight[2][2][6]: -6.167995  
weight[2][3][0]: 6.477601  
weight[2][3][1]: -2.801642  
weight[2][3][2]: -5.056403  
weight[2][3][3]: -1.796334  
weight[2][3][4]: 8.799793  
weight[2][3][5]: -7.487347  
weight[2][3][6]: 2.271102  
weight[2][4][0]: -8.070490  
weight[2][4][1]: 7.384803  
weight[2][4][2]: -4.679778  
weight[2][4][3]: 6.021894  
weight[2][4][4]: 8.341533  
weight[2][4][5]: 5.577064  
weight[2][4][6]: 0.530860  
weight[2][5][0]: 6.654666  
weight[2][5][1]: 3.554341  
weight[2][5][2]: -4.785960  
weight[2][5][3]: 8.611599  
weight[2][5][4]: 6.329325  
weight[2][5][5]: -5.545141  
weight[2][5][6]: 1.633716  
weight[2][6][0]: -0.799187  
weight[2][6][1]: -1.714016  
weight[2][6][2]: -3.236910  
weight[2][6][3]: -0.205491  
weight[2][6][4]: 4.459275  
weight[2][6][5]: 5.153455  
weight[2][6][6]: -3.643570  
weight[3][0][0]: 3.063407  
weight[3][0][1]: 5.497949  
weight[3][0][2]: -5.378568  
weight[3][0][3]: -7.365783  
weight[3][0][4]: -1.382810  
weight[3][0][5]: 8.539319  
weight[3][0][6]: -8.468186  
network\_fuzzy\_number: 0.460000

[ FEED RATE DATA\_FILE : END MILLING, DATA\_GROUP : B ]

current error : 99.173902  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.689655 2.500000 2.500000 0.062278  
scale: 18.603449 1.950000 1.950000 0.075761  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: -2.845733  
weight[1][0][1]: -7.653131  
weight[1][0][2]: -1.717733  
weight[1][1][0]: -6.098122  
weight[1][1][1]: 2.192474  
weight[1][1][2]: 12.499296  
weight[1][2][0]: 7.665620  
weight[1][2][1]: -10.609261  
weight[1][2][2]: 3.808000  
weight[1][3][0]: 2.916615  
weight[1][3][1]: -17.976135  
weight[1][3][2]: 4.481600  
weight[1][4][0]: -1.152273  
weight[1][4][1]: -7.314234  
weight[1][4][2]: -1.281512  
weight[1][5][0]: -5.284066  
weight[1][5][1]: -2.956699  
weight[1][5][2]: -6.769166  
weight[1][6][0]: 10.807280  
weight[1][6][1]: 5.485594  
weight[1][6][2]: 6.211566  
weight[2][0][0]: 3.994008  
weight[2][0][1]: 4.660713  
weight[2][0][2]: 4.717989  
weight[2][0][3]: 6.914041  
weight[2][0][4]: -8.377004  
weight[2][0][5]: 11.061344  
weight[2][0][6]: -3.178822  
weight[2][1][0]: -6.986627  
weight[2][1][1]: -6.878841  
weight[2][1][2]: -4.986726  
weight[2][1][3]: -7.225972  
weight[2][1][4]: -7.012719  
weight[2][1][5]: 4.126753  
weight[2][1][6]: -5.153362  
weight[2][2][0]: -4.225441  
weight[2][2][1]: 2.178913  
weight[2][2][2]: -3.617675  
weight[2][2][3]: -1.086817  
weight[2][2][4]: -1.533616  
weight[2][2][5]: 2.841231  
weight[2][2][6]: -6.167675  
weight[2][3][0]: 6.477881  
weight[2][3][1]: -2.800802  
weight[2][3][2]: -5.056143  
weight[2][3][3]: -1.796014  
weight[2][3][4]: 8.800073  
weight[2][3][5]: -7.487567  
weight[2][3][6]: 2.271122  
weight[2][4][0]: -8.070950  
weight[2][4][1]: 7.383803  
weight[2][4][2]: -4.678878  
weight[2][4][3]: 6.021134  
weight[2][4][4]: 8.340693  
weight[2][4][5]: 5.577384  
weight[2][4][6]: 0.530800  
weight[2][5][0]: 6.654506  
weight[2][5][1]: 3.554821  
weight[2][5][2]: -4.785580  
weight[2][5][3]: 8.612379  
weight[2][5][4]: 6.329985  
weight[2][5][5]: -5.544821  
weight[2][5][6]: 1.633536  
weight[2][6][0]: -0.798387  
weight[2][6][1]: -1.713456  
weight[2][6][2]: -3.236610  
weight[2][6][3]: -0.204911  
weight[2][6][4]: 4.460075  
weight[2][6][5]: 5.153115  
weight[2][6][6]: -3.643510  
weight[3][0][0]: 3.062987  
weight[3][0][1]: 5.498649  
weight[3][0][2]: -5.379128  
weight[3][0][3]: -7.366123  
weight[3][0][4]: -1.383070  
weight[3][0][5]: 8.539039  
weight[3][0][6]: -8.467826  
network\_fuzzy\_number: 0.600000

[ VELOCITY DATA\_FILE : END MILLING, DATA\_GROUP : A ]

current error : 136.050070  
 SUCCESS in LEARNING WEIGHT at 3 iteration

```

num_layer: 4
num_node: 3 7 7 1
average: 15.689655 2.500000 2.500000 30.329741
scale: 18.603449 1.950000 1.950000 34.228664
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: 0.497918
weight[1][0][1]: -13.276460
weight[1][0][2]: 7.120480
weight[1][1][0]: 0.336773
weight[1][1][1]: -1.004199
weight[1][1][2]: 6.013226
weight[1][2][0]: 2.079655
weight[1][2][1]: -9.084300
weight[1][2][2]: -0.244252
weight[1][3][0]: 10.328793
weight[1][3][1]: -4.176925
weight[1][3][2]: -1.216606
weight[1][4][0]: -7.752983
weight[1][4][1]: 1.935308
weight[1][4][2]: 3.740261
weight[1][5][0]: 0.453248
weight[1][5][1]: 2.810408
weight[1][5][2]: -5.832337
weight[1][6][0]: -0.043118
weight[1][6][1]: 7.673600
weight[1][6][2]: 6.455915
weight[2][0][0]: 7.596960
weight[2][0][1]: -5.871951
weight[2][0][2]: 7.782647
weight[2][0][3]: 7.379719
weight[2][0][4]: 3.927802
weight[2][0][5]: -1.877460
weight[2][0][6]: 3.842272
weight[2][1][0]: 1.277552
weight[2][1][1]: -0.746348
weight[2][1][2]: 3.378707
weight[2][1][3]: -1.193531
weight[2][1][4]: -8.487616
weight[2][1][5]: -5.605089
weight[2][1][6]: -10.566188
weight[2][2][0]: -0.348115
weight[2][2][1]: 6.555601
weight[2][2][2]: 8.726664
weight[2][2][3]: 3.467679
weight[2][2][4]: 1.638961
weight[2][2][5]: -6.373739
weight[2][2][6]: -5.465515
weight[2][3][0]: 0.252757
weight[2][3][1]: -9.057624
weight[2][3][2]: -4.776770
weight[2][3][3]: -3.591695
weight[2][3][4]: 9.065203
weight[2][3][5]: -2.593729
weight[2][3][6]: 1.778423
weight[2][4][0]: -3.788075
weight[2][4][1]: 1.063499
weight[2][4][2]: -6.899496
weight[2][4][3]: -5.696107
weight[2][4][4]: 2.278041
weight[2][4][5]: 6.751996
weight[2][4][6]: -2.189422
weight[2][5][0]: -6.550314
weight[2][5][1]: 2.874353
weight[2][5][2]: 2.768208
weight[2][5][3]: 1.511434
weight[2][5][4]: -6.703439
weight[2][5][5]: -4.415193
weight[2][5][6]: 4.265379
weight[2][6][0]: 2.893063
weight[2][6][1]: -2.922775
weight[2][6][2]: -7.750795
weight[2][6][3]: -1.235944
weight[2][6][4]: -7.421713
weight[2][6][5]: -4.602150
weight[2][6][6]: -0.935582
weight[3][0][0]: 9.256870
weight[3][0][1]: 3.206818
weight[3][0][2]: -8.453716
weight[3][0][3]: 0.341967
weight[3][0][4]: -5.050214
weight[3][0][5]: 4.231766
weight[3][0][6]: 6.894534
network_fuzzy_number: 0.460000

```



[ VELOCITY DATA\_FILE : END MILLING, DATA\_GROUP : B ]

current error : 121.356662  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.689655 2.500000 2.500000 30.329741  
scale: 18.603449 1.950000 1.950000 34.228664  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: 0.497518  
weight[1][0][1]: -13.275820  
weight[1][0][2]: 7.121280  
weight[1][1][0]: 0.336893  
weight[1][1][1]: -1.004859  
weight[1][1][2]: 6.014126  
weight[1][2][0]: 2.078955  
weight[1][2][1]: -9.084340  
weight[1][2][2]: -0.244732  
weight[1][3][0]: 10.327873  
weight[1][3][1]: -4.176765  
weight[1][3][2]: -1.216186  
weight[1][4][0]: -7.752403  
weight[1][4][1]: 1.936148  
weight[1][4][2]: 3.740461  
weight[1][5][0]: 0.452488  
weight[1][5][1]: 2.809828  
weight[1][5][2]: -5.832077  
weight[1][6][0]: -0.043178  
weight[1][6][1]: 7.672980  
weight[1][6][2]: 6.455735  
weight[2][0][0]: 7.597760  
weight[2][0][1]: -5.871251  
weight[2][0][2]: 7.781927  
weight[2][0][3]: 7.378899  
weight[2][0][4]: 3.927842  
weight[2][0][5]: -1.877040  
weight[2][0][6]: 3.842852  
weight[2][1][0]: 1.276872  
weight[2][1][1]: -0.745728  
weight[2][1][2]: 3.378727  
weight[2][1][3]: -1.192631  
weight[2][1][4]: -8.486756  
weight[2][1][5]: -5.605409  
weight[2][1][6]: -10.566988  
weight[2][2][0]: -0.347535  
weight[2][2][1]: 6.556501  
weight[2][2][2]: 8.726884  
weight[2][2][3]: 3.468519  
weight[2][2][4]: 1.639741  
weight[2][2][5]: -6.372979  
weight[2][2][6]: -5.465195  
weight[2][3][0]: 0.253037  
weight[2][3][1]: -9.056784  
weight[2][3][2]: -4.776510  
weight[2][3][3]: -3.591375  
weight[2][3][4]: 9.065483  
weight[2][3][5]: -2.593949  
weight[2][3][6]: 1.778443  
weight[2][4][0]: -3.788535  
weight[2][4][1]: 1.062499  
weight[2][4][2]: -6.898596  
weight[2][4][3]: -5.696867  
weight[2][4][4]: 2.277201  
weight[2][4][5]: 6.752316  
weight[2][4][6]: -2.189482  
weight[2][5][0]: -6.550474  
weight[2][5][1]: 2.874833  
weight[2][5][2]: 2.768588  
weight[2][5][3]: 1.512214  
weight[2][5][4]: -6.702779  
weight[2][5][5]: -4.414873  
weight[2][5][6]: 4.265199  
weight[2][6][0]: 2.893863  
weight[2][6][1]: -2.922215  
weight[2][6][2]: -7.750495  
weight[2][6][3]: -1.235364  
weight[2][6][4]: -7.420913  
weight[2][6][5]: -4.602490  
weight[2][6][6]: -0.935522  
weight[3][0][0]: 9.256450  
weight[3][0][1]: 3.207518  
weight[3][0][2]: -8.454276  
weight[3][0][3]: 0.341627  
weight[3][0][4]: -5.050474  
weight[3][0][5]: 4.231486  
weight[3][0][6]: 6.894894  
network\_fuzzy\_number: 0.600000

[ FEED RATE DATA\_FILE : SLOTTING, DATA\_GROUP : A ]

current error : 168.053655  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.505241 2.490566 2.509434 0.048281  
scale: 18.843185 1.937736 1.937736 0.049765  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: 2.997760  
weight[1][0][1]: -5.608577  
weight[1][0][2]: -0.980565  
weight[1][1][0]: -1.437726  
weight[1][1][1]: 6.597458  
weight[1][1][2]: 13.100651  
weight[1][2][0]: 7.766527  
weight[1][2][1]: -24.631558  
weight[1][2][2]: 9.349022  
weight[1][3][0]: 4.711905  
weight[1][3][1]: -6.182150  
weight[1][3][2]: 7.552678  
weight[1][4][0]: -1.862104  
weight[1][4][1]: -4.811003  
weight[1][4][2]: 0.103523  
weight[1][5][0]: 17.265811  
weight[1][5][1]: -12.156156  
weight[1][5][2]: -7.520745  
weight[1][6][0]: 11.568875  
weight[1][6][1]: 5.665075  
weight[1][6][2]: 11.903053  
weight[2][0][0]: 2.505719  
weight[2][0][1]: -2.235828  
weight[2][0][2]: 7.237868  
weight[2][0][3]: 4.801330  
weight[2][0][4]: -9.893105  
weight[2][0][5]: 8.416686  
weight[2][0][6]: -1.218563  
weight[2][1][0]: -10.593925  
weight[2][1][1]: -9.968582  
weight[2][1][2]: -4.251204  
weight[2][1][3]: 5.877573  
weight[2][1][4]: -14.203095  
weight[2][1][5]: 3.391655  
weight[2][1][6]: -1.225304  
weight[2][2][0]: -1.677674  
weight[2][2][1]: 2.306317  
weight[2][2][2]: -4.320802  
weight[2][2][3]: -0.630168  
weight[2][2][4]: -4.105729  
weight[2][2][5]: 3.759140  
weight[2][2][6]: -4.627534  
weight[2][3][0]: 7.728450  
weight[2][3][1]: -3.952348  
weight[2][3][2]: -5.219000  
weight[2][3][3]: -1.104989  
weight[2][3][4]: 8.776885  
weight[2][3][5]: -7.569434  
weight[2][3][6]: 1.772897  
weight[2][4][0]: -8.121376  
weight[2][4][1]: 13.103450  
weight[2][4][2]: -2.446780  
weight[2][4][3]: 11.793358  
weight[2][4][4]: 8.540354  
weight[2][4][5]: 4.851355  
weight[2][4][6]: -1.381343  
weight[2][5][0]: 4.855192  
weight[2][5][1]: 9.720039  
weight[2][5][2]: 7.609705  
weight[2][5][3]: 10.052203  
weight[2][5][4]: 4.026074  
weight[2][5][5]: -7.434603  
weight[2][5][6]: 26.601804  
weight[2][6][0]: -1.179309  
weight[2][6][1]: -0.313021  
weight[2][6][2]: -3.967782  
weight[2][6][3]: -0.014664  
weight[2][6][4]: 4.790326  
weight[2][6][5]: 5.734912  
weight[2][6][6]: -3.763964  
weight[3][0][0]: 3.606735  
weight[3][0][1]: 6.471830  
weight[3][0][2]: -4.009987  
weight[3][0][3]: -6.683193  
weight[3][0][4]: -1.000297  
weight[3][0][5]: 8.522733  
weight[3][0][6]: -8.370634  
network\_fuzzy\_number: 0.460000



[ FEED RATE DATA\_FILE : SLOTTING, DATA\_GROUP : B ]

current error : 149.903860  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.505241 2.490566 2.509434 0.048281  
scale: 18.843185 1.937736 1.937736 0.049765  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: 2.997360  
weight[1][0][1]: -5.607937  
weight[1][0][2]: -0.979765  
weight[1][1][0]: -1.437606  
weight[1][1][1]: 6.596798  
weight[1][1][2]: 13.101551  
weight[1][2][0]: 7.765827  
weight[1][2][1]: -24.631598  
weight[1][2][2]: 9.348542  
weight[1][3][0]: 4.710985  
weight[1][3][1]: -6.181990  
weight[1][3][2]: 7.553098  
weight[1][4][0]: -1.861524  
weight[1][4][1]: -4.810163  
weight[1][4][2]: 0.103723  
weight[1][5][0]: 17.265051  
weight[1][5][1]: -12.156736  
weight[1][5][2]: -7.520485  
weight[1][6][0]: 11.568815  
weight[1][6][1]: 5.664455  
weight[1][6][2]: 11.902873  
weight[2][0][0]: 2.506519  
weight[2][0][1]: -2.235128  
weight[2][0][2]: 7.237148  
weight[2][0][3]: 4.800510  
weight[2][0][4]: -9.893065  
weight[2][0][5]: 8.417106  
weight[2][0][6]: -1.217983  
weight[2][1][0]: -10.594605  
weight[2][1][1]: -9.967962  
weight[2][1][2]: -4.251184  
weight[2][1][3]: 5.878473  
weight[2][1][4]: -14.202235  
weight[2][1][5]: 3.391335  
weight[2][1][6]: -1.226104  
weight[2][2][0]: -1.677094  
weight[2][2][1]: 2.307217  
weight[2][2][2]: -4.320582  
weight[2][2][3]: -0.629328  
weight[2][2][4]: -4.104949  
weight[2][2][5]: 3.759900  
weight[2][2][6]: -4.627214  
weight[2][3][0]: 7.728730  
weight[2][3][1]: -3.951508  
weight[2][3][2]: -5.218740  
weight[2][3][3]: -1.104669  
weight[2][3][4]: 8.777165  
weight[2][3][5]: -7.569654  
weight[2][3][6]: 1.772917  
weight[2][4][0]: -8.121836  
weight[2][4][1]: 13.102450  
weight[2][4][2]: -2.445880  
weight[2][4][3]: 11.792598  
weight[2][4][4]: 8.539514  
weight[2][4][5]: 4.851675  
weight[2][4][6]: -1.381403  
weight[2][5][0]: 4.855032  
weight[2][5][1]: 9.720519  
weight[2][5][2]: 7.610085  
weight[2][5][3]: 10.052983  
weight[2][5][4]: 4.026734  
weight[2][5][5]: -7.434283  
weight[2][5][6]: 26.601624  
weight[2][6][0]: -1.178509  
weight[2][6][1]: -0.312461  
weight[2][6][2]: -3.967482  
weight[2][6][3]: -0.014084  
weight[2][6][4]: 4.791126  
weight[2][6][5]: 5.734572  
weight[2][6][6]: -3.763904  
weight[3][0][0]: 3.606315  
weight[3][0][1]: 6.472530  
weight[3][0][2]: -4.010547  
weight[3][0][3]: -6.683533  
weight[3][0][4]: -1.000557  
weight[3][0][5]: 8.522453  
weight[3][0][6]: -8.370274  
network\_fuzzy\_number: 0.600000

[ VELOCITY DATA\_FILE : SLOTTING, DATA\_GROUP : A ]

current error : 137.421856  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 3 7 7 1  
average: 15.505241 2.490566 2.509434 20.530397  
scale: 18.843185 1.937736 1.937736 24.089518  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: 1.121947  
weight[1][0][1]: -13.050166  
weight[1][0][2]: 7.480469  
weight[1][1][0]: 0.471120  
weight[1][1][1]: -0.704228  
weight[1][1][2]: 6.727654  
weight[1][2][0]: 3.013723  
weight[1][2][1]: -9.489977  
weight[1][2][2]: -0.628942  
weight[1][3][0]: 7.776756  
weight[1][3][1]: -3.233731  
weight[1][3][2]: -1.969897  
weight[1][4][0]: -12.279926  
weight[1][4][1]: 1.616722  
weight[1][4][2]: 3.511426  
weight[1][5][0]: 1.643284  
weight[1][5][1]: 1.512687  
weight[1][5][2]: -6.345471  
weight[1][6][0]: 1.431102  
weight[1][6][1]: 7.573383  
weight[1][6][2]: 13.681547  
weight[2][0][0]: 8.087471  
weight[2][0][1]: -5.887078  
weight[2][0][2]: 8.059729  
weight[2][0][3]: 8.256450  
weight[2][0][4]: 3.066448  
weight[2][0][5]: -1.736816  
weight[2][0][6]: 3.620983  
weight[2][1][0]: 0.572068  
weight[2][1][1]: -1.910757  
weight[2][1][2]: 3.232567  
weight[2][1][3]: -1.428942  
weight[2][1][4]: -0.647500  
weight[2][1][5]: -0.730068  
weight[2][1][6]: -8.080822  
weight[2][2][0]: 1.045830  
weight[2][2][1]: 7.067427  
weight[2][2][2]: 11.584803  
weight[2][2][3]: 5.361922  
weight[2][2][4]: 1.542867  
weight[2][2][5]: -8.053378  
weight[2][2][6]: -4.573818  
weight[2][3][0]: -1.902337  
weight[2][3][1]: -8.607800  
weight[2][3][2]: -7.365463  
weight[2][3][3]: -2.189472  
weight[2][3][4]: 10.272665  
weight[2][3][5]: -2.263466  
weight[2][3][6]: 2.371417  
weight[2][4][0]: -1.699960  
weight[2][4][1]: 1.752254  
weight[2][4][2]: -8.501002  
weight[2][4][3]: -4.868737  
weight[2][4][4]: 0.948659  
weight[2][4][5]: 5.620731  
weight[2][4][6]: -1.862915  
weight[2][5][0]: -8.108773  
weight[2][5][1]: 3.252346  
weight[2][5][2]: 2.674249  
weight[2][5][3]: 1.354666  
weight[2][5][4]: -5.812587  
weight[2][5][5]: -2.773284  
weight[2][5][6]: 4.442303  
weight[2][6][0]: 4.109054  
weight[2][6][1]: -3.543769  
weight[2][6][2]: -9.634303  
weight[2][6][3]: -1.980979  
weight[2][6][4]: -18.293858  
weight[2][6][5]: -9.984095  
weight[2][6][6]: 0.460871  
weight[3][0][0]: 9.222344  
weight[3][0][1]: 3.029992  
weight[3][0][2]: -8.588248  
weight[3][0][3]: 0.847649  
weight[3][0][4]: -5.045895  
weight[3][0][5]: 4.171235  
weight[3][0][6]: 6.571701  
network\_fuzzy\_number: 0.460000

[ VELOCITY DATA\_FILE : SLOTTING, DATA\_GROUP : B ]

current error : 122.580296  
 SUCCESS in LEARNING WEIGHT at 3 iteration

```

num_layer: 4
num_node: 3 7 7 1
average: 15.505241 2.490566 2.509434 20.530397
scale: 18.843185 1.937736 1.937736 24.089518
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: 1.121547
weight[1][0][1]: -13.049526
weight[1][0][2]: 7.481269
weight[1][1][0]: 0.471240
weight[1][1][1]: -0.704888
weight[1][1][2]: 6.728554
weight[1][2][0]: 3.013023
weight[1][2][1]: -9.490017
weight[1][2][2]: -0.629422
weight[1][3][0]: 7.775836
weight[1][3][1]: -3.233571
weight[1][3][2]: -1.969477
weight[1][4][0]: -12.279346
weight[1][4][1]: 1.617562
weight[1][4][2]: 3.511626
weight[1][5][0]: 1.642524
weight[1][5][1]: 1.512107
weight[1][5][2]: -6.345211
weight[1][6][0]: 1.431042
weight[1][6][1]: 7.572763
weight[1][6][2]: 13.681367
weight[2][0][0]: 8.088271
weight[2][0][1]: -5.886378
weight[2][0][2]: 8.059009
weight[2][0][3]: 8.255630
weight[2][0][4]: 3.066488
weight[2][0][5]: -1.736396
weight[2][0][6]: 3.621563
weight[2][1][0]: 0.571388
weight[2][1][1]: -1.910137
weight[2][1][2]: 3.232587
weight[2][1][3]: -1.428042
weight[2][1][4]: -0.646640
weight[2][1][5]: -0.730388
weight[2][1][6]: -8.081622
weight[2][2][0]: 1.046410
weight[2][2][1]: 7.068327
weight[2][2][2]: 11.585023
weight[2][2][3]: 5.362762
weight[2][2][4]: 1.543647
weight[2][2][5]: -8.052618
weight[2][2][6]: -4.573498
weight[2][3][0]: -1.902057
weight[2][3][1]: -8.606960
weight[2][3][2]: -7.365203
weight[2][3][3]: -2.189152
weight[2][3][4]: 10.272945
weight[2][3][5]: -2.263686
weight[2][3][6]: 2.371437
weight[2][4][0]: -1.700420
weight[2][4][1]: 1.751254
weight[2][4][2]: -8.500102
weight[2][4][3]: -4.869497
weight[2][4][4]: 0.947819
weight[2][4][5]: 5.621051
weight[2][4][6]: -1.862975
weight[2][5][0]: -8.108933
weight[2][5][1]: 3.252826
weight[2][5][2]: 2.674629
weight[2][5][3]: 1.355446
weight[2][5][4]: -5.811927
weight[2][5][5]: -2.772964
weight[2][5][6]: 4.442123
weight[2][6][0]: 4.109854
weight[2][6][1]: -3.543209
weight[2][6][2]: -9.634003
weight[2][6][3]: -1.980399
weight[2][6][4]: -18.293058
weight[2][6][5]: -9.984435
weight[2][6][6]: 0.460931
weight[3][0][0]: 9.221924
weight[3][0][1]: 3.030692
weight[3][0][2]: -8.588808
weight[3][0][3]: 0.847309
weight[3][0][4]: -5.046155
weight[3][0][5]: 4.170955
weight[3][0][6]: 6.572061
network_fuzzy_number: 0.600000

```

[ FEED RATE DATA\_FILE : DRILLING, DATA GROUP : A 1

current error : 53.522277  
 SUCCESS in LEARNING WEIGHT at 3 iteration

```

num_layer: 4
num_node: 2 7 7 1
average: 15.500000 4.500000 0.216813
scale: 18.850000 4.550000 0.268856
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: -4.414268
weight[1][0][1]: -2.157145
weight[1][1][0]: -4.190178
weight[1][1][1]: 9.825325
weight[1][2][0]: -6.315767
weight[1][2][1]: -0.117348
weight[1][3][0]: 5.306809
weight[1][3][1]: 0.006907
weight[1][4][0]: -2.755841
weight[1][4][1]: -1.699554
weight[1][5][0]: 0.471963
weight[1][5][1]: -2.695948
weight[1][6][0]: -2.296362
weight[1][6][1]: 0.085041
weight[2][0][0]: 8.709914
weight[2][0][1]: 2.749334
weight[2][0][2]: -10.078569
weight[2][0][3]: -1.999356
weight[2][0][4]: -5.736184
weight[2][0][5]: 4.495504
weight[2][0][6]: 6.221989
weight[2][1][0]: -8.921888
weight[2][1][1]: -6.235290
weight[2][1][2]: 2.090903
weight[2][1][3]: -6.298052
weight[2][1][4]: 5.348967
weight[2][1][5]: 5.855761
weight[2][1][6]: 6.315894
weight[2][2][0]: -0.844053
weight[2][2][1]: 13.182526
weight[2][2][2]: -2.006908
weight[2][2][3]: 6.685758
weight[2][2][4]: -1.942516
weight[2][2][5]: 0.176997
weight[2][2][6]: 0.862346
weight[2][3][0]: 6.359332
weight[2][3][1]: 7.856357
weight[2][3][2]: -5.467706
weight[2][3][3]: -4.218781
weight[2][3][4]: 0.381501
weight[2][3][5]: 9.352487
weight[2][3][6]: 4.614012
weight[2][4][0]: 2.327744
weight[2][4][1]: 3.752383
weight[2][4][2]: 8.975348
weight[2][4][3]: -2.525395
weight[2][4][4]: -1.005725
weight[2][4][5]: 4.407702
weight[2][4][6]: 6.467439
weight[2][5][0]: -7.496770
weight[2][5][1]: 1.815832
weight[2][5][2]: 5.168572
weight[2][5][3]: 8.862790
weight[2][5][4]: -5.823097
weight[2][5][5]: 2.134800
weight[2][5][6]: -10.120739
weight[2][6][0]: 2.957736
weight[2][6][1]: 2.301267
weight[2][6][2]: -16.450649
weight[2][6][3]: -2.364281
weight[2][6][4]: -5.425957
weight[2][6][5]: -4.929163
weight[2][6][6]: -5.801041
weight[3][0][0]: -1.437082
weight[3][0][1]: -6.939850
weight[3][0][2]: 7.586436
weight[3][0][3]: 3.009584
weight[3][0][4]: -5.190484
weight[3][0][5]: 3.138463
weight[3][0][6]: 3.790506
network_fuzzy_number: 0.460000
  
```



[ FEED RATE DATA\_FILE : DRILLING, DATA\_GROUP : B ]

current error : 47.741871  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 2 7 7 1  
average: 15.500000 4.500000 0.216813  
scale: 18.850000 4.550000 0.268856  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: -4.414668  
weight[1][0][1]: -2.156505  
weight[1][1][0]: -4.189378  
weight[1][1][1]: 9.825445  
weight[1][2][0]: -6.316427  
weight[1][2][1]: -0.116448  
weight[1][3][0]: 5.306109  
weight[1][3][1]: 0.006867  
weight[1][4][0]: -2.756321  
weight[1][4][1]: -1.700474  
weight[1][5][0]: 0.472123  
weight[1][5][1]: -2.695528  
weight[1][6][0]: -2.295782  
weight[1][6][1]: 0.085881  
weight[2][0][0]: 8.710114  
weight[2][0][1]: 2.748574  
weight[2][0][2]: -10.079149  
weight[2][0][3]: -1.999096  
weight[2][0][4]: -5.736244  
weight[2][0][5]: 4.494884  
weight[2][0][6]: 6.221809  
weight[2][1][0]: -8.921088  
weight[2][1][1]: -6.234590  
weight[2][1][2]: 2.090183  
weight[2][1][3]: -6.298872  
weight[2][1][4]: 5.349007  
weight[2][1][5]: 5.856181  
weight[2][1][6]: 6.316474  
weight[2][2][0]: -0.844733  
weight[2][2][1]: 13.183146  
weight[2][2][2]: -2.006888  
weight[2][2][3]: 6.686658  
weight[2][2][4]: -1.941656  
weight[2][2][5]: 0.176677  
weight[2][2][6]: 0.861546  
weight[2][3][0]: 6.359912  
weight[2][3][1]: 7.857257  
weight[2][3][2]: -5.467486  
weight[2][3][3]: -4.217941  
weight[2][3][4]: 0.382281  
weight[2][3][5]: 9.353247  
weight[2][3][6]: 4.614332  
weight[2][4][0]: 2.328024  
weight[2][4][1]: 3.753223  
weight[2][4][2]: 8.975608  
weight[2][4][3]: -2.525075  
weight[2][4][4]: -1.005445  
weight[2][4][5]: 4.407482  
weight[2][4][6]: 6.467459  
weight[2][5][0]: -7.497230  
weight[2][5][1]: 1.814832  
weight[2][5][2]: 5.169472  
weight[2][5][3]: 8.862030  
weight[2][5][4]: -5.823937  
weight[2][5][5]: 2.135120  
weight[2][5][6]: -10.120799  
weight[2][6][0]: 2.957576  
weight[2][6][1]: 2.301747  
weight[2][6][2]: -16.450269  
weight[2][6][3]: -2.363501  
weight[2][6][4]: -5.425297  
weight[2][6][5]: -4.928843  
weight[2][6][6]: -5.801221  
weight[3][0][0]: -1.436282  
weight[3][0][1]: -6.939290  
weight[3][0][2]: 7.586736  
weight[3][0][3]: 3.010164  
weight[3][0][4]: -5.189684  
weight[3][0][5]: 3.138123  
weight[3][0][6]: 3.790566  
network\_fuzzy\_number: 0.290000

[ VELOCITY DATA\_FILE : DRILLING, DATA\_GROUP : A ]

current error : 54.024810  
 SUCCESS in LEARNING WEIGHT at 3 iteration

```

num_layer: 4
num_node: 2 7 7 1
average: 15.500000 4.500000 17.691668
scale: 18.850000 4.550000 17.799168
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: -3.765914
weight[1][0][1]: 3.262149
weight[1][1][0]: 2.476929
weight[1][1][1]: -2.001617
weight[1][2][0]: 9.461788
weight[1][2][1]: -6.881207
weight[1][3][0]: -2.529431
weight[1][3][1]: -8.320778
weight[1][4][0]: 3.684689
weight[1][4][1]: 0.449146
weight[1][5][0]: 10.651573
weight[1][5][1]: -3.766667
weight[1][6][0]: 2.864561
weight[1][6][1]: -6.737730
weight[2][0][0]: 0.986774
weight[2][0][1]: -4.232764
weight[2][0][2]: 8.844917
weight[2][0][3]: 8.620640
weight[2][0][4]: -8.353075
weight[2][0][5]: 1.977288
weight[2][0][6]: 3.779506
weight[2][1][0]: 0.844297
weight[2][1][1]: -2.631861
weight[2][1][2]: -0.578450
weight[2][1][3]: 2.190437
weight[2][1][4]: -7.328485
weight[2][1][5]: 0.187825
weight[2][1][6]: -2.897034
weight[2][2][0]: 7.927042
weight[2][2][1]: -6.140959
weight[2][2][2]: -10.321299
weight[2][2][3]: 6.207748
weight[2][2][4]: 7.207162
weight[2][2][5]: 1.193921
weight[2][2][6]: 7.396786
weight[2][3][0]: -3.694670
weight[2][3][1]: 0.940500
weight[2][3][2]: 9.833953
weight[2][3][3]: 7.920642
weight[2][3][4]: 1.914710
weight[2][3][5]: -7.823556
weight[2][3][6]: 6.548625
weight[2][4][0]: -8.627557
weight[2][4][1]: -8.128078
weight[2][4][2]: 0.772700
weight[2][4][3]: 6.758789
weight[2][4][4]: -8.128588
weight[2][4][5]: -7.793280
weight[2][4][6]: 3.603700
weight[2][5][0]: 9.126287
weight[2][5][1]: 7.400914
weight[2][5][2]: 4.047023
weight[2][5][3]: -1.015688
weight[2][5][4]: 6.967173
weight[2][5][5]: 0.839354
weight[2][5][6]: -4.566492
weight[2][6][0]: 2.504280
weight[2][6][1]: -1.161939
weight[2][6][2]: 6.430199
weight[2][6][3]: -5.452137
weight[2][6][4]: -7.725280
weight[2][6][5]: -3.865673
weight[2][6][6]: -1.336447
weight[3][0][0]: 0.180575
weight[3][0][1]: -4.760944
weight[3][0][2]: -7.996119
weight[3][0][3]: -0.252181
weight[3][0][4]: -4.402680
weight[3][0][5]: -0.395733
weight[3][0][6]: 1.784837
network_fuzzy_number: 0.460000
  
```



[ VELOCITY DATA\_FILE : DRILLING, DATA\_GROUP : B ]

current error : 48.190131  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 2 7 7 1  
average: 15.500000 4.500000 17.691668  
scale: 18.850000 4.550000 17.799168  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: -3.766314  
weight[1][0][1]: 3.262789  
weight[1][1][0]: 2.477729  
weight[1][1][1]: -2.001497  
weight[1][2][0]: 9.461128  
weight[1][2][1]: -6.880307  
weight[1][3][0]: -2.530131  
weight[1][3][1]: -8.320818  
weight[1][4][0]: 3.684209  
weight[1][4][1]: 0.448226  
weight[1][5][0]: 10.651733  
weight[1][5][1]: -3.766247  
weight[1][6][0]: 2.865141  
weight[1][6][1]: -6.736890  
weight[2][0][0]: 0.986974  
weight[2][0][1]: -4.233524  
weight[2][0][2]: 8.844337  
weight[2][0][3]: 8.620900  
weight[2][0][4]: -8.353135  
weight[2][0][5]: 1.976668  
weight[2][0][6]: 3.779326  
weight[2][1][0]: 0.845097  
weight[2][1][1]: -2.631161  
weight[2][1][2]: -0.579170  
weight[2][1][3]: 2.189617  
weight[2][1][4]: -7.328445  
weight[2][1][5]: 0.188245  
weight[2][1][6]: -2.896454  
weight[2][2][0]: 7.926362  
weight[2][2][1]: -6.140339  
weight[2][2][2]: -10.321279  
weight[2][2][3]: 6.208648  
weight[2][2][4]: 7.208022  
weight[2][2][5]: 1.193601  
weight[2][2][6]: 7.395986  
weight[2][3][0]: -3.694090  
weight[2][3][1]: 0.941400  
weight[2][3][2]: 9.834173  
weight[2][3][3]: 7.921482  
weight[2][3][4]: 1.915490  
weight[2][3][5]: -7.822796  
weight[2][3][6]: 6.548945  
weight[2][4][0]: -8.627277  
weight[2][4][1]: -8.127238  
weight[2][4][2]: 0.772960  
weight[2][4][3]: 6.759109  
weight[2][4][4]: -8.128308  
weight[2][4][5]: -7.793500  
weight[2][4][6]: 3.603720  
weight[2][5][0]: 9.125827  
weight[2][5][1]: 7.399914  
weight[2][5][2]: 4.047923  
weight[2][5][3]: -1.016448  
weight[2][5][4]: 6.966333  
weight[2][5][5]: 0.839674  
weight[2][5][6]: -4.566552  
weight[2][6][0]: 2.504120  
weight[2][6][1]: -1.161459  
weight[2][6][2]: 6.430579  
weight[2][6][3]: -5.451357  
weight[2][6][4]: -7.724620  
weight[2][6][5]: -3.865353  
weight[2][6][6]: -1.336627  
weight[3][0][0]: 0.181375  
weight[3][0][1]: -4.760384  
weight[3][0][2]: -7.995819  
weight[3][0][3]: -0.251601  
weight[3][0][4]: -4.401880  
weight[3][0][5]: -0.396073  
weight[3][0][6]: 1.784897  
network\_fuzzy\_number: 0.290000

[ VELOCITY DATA\_FILE : TAPPING, DATA\_GROUP : A ]

current error : 31.511653  
SUCCESS in LEARNING WEIGHT at 3 iteration

num\_layer: 4  
num\_node: 2 7 7 1  
average: 15.500000 2.500000 8.786667  
scale: 18.850000 1.950000 10.642667  
threshold: 1.000000  
beta: 1.000000  
theta: 1.000000  
weight[1][0][0]: 6.590702  
weight[1][0][1]: -5.551657  
weight[1][1][0]: 10.385815  
weight[1][1][1]: -4.615048  
weight[1][2][0]: 5.831912  
weight[1][2][1]: 4.820504  
weight[1][3][0]: -5.453154  
weight[1][3][1]: -7.403116  
weight[1][4][0]: -5.032429  
weight[1][4][1]: -10.358028  
weight[1][5][0]: 2.967065  
weight[1][5][1]: 7.870335  
weight[1][6][0]: 3.481801  
weight[1][6][1]: -0.565490  
weight[2][0][0]: -2.315511  
weight[2][0][1]: 3.769059  
weight[2][0][2]: 4.592540  
weight[2][0][3]: -8.096348  
weight[2][0][4]: 2.524933  
weight[2][0][5]: -11.743232  
weight[2][0][6]: 1.615258  
weight[2][1][0]: -5.986881  
weight[2][1][1]: 5.021368  
weight[2][1][2]: -9.311251  
weight[2][1][3]: -2.393033  
weight[2][1][4]: 5.028484  
weight[2][1][5]: 9.885436  
weight[2][1][6]: -2.785423  
weight[2][2][0]: -2.296504  
weight[2][2][1]: 6.429546  
weight[2][2][2]: -0.353251  
weight[2][2][3]: 5.928395  
weight[2][2][4]: -2.272628  
weight[2][2][5]: 0.134091  
weight[2][2][6]: 6.574427  
weight[2][3][0]: -2.609218  
weight[2][3][1]: 2.660218  
weight[2][3][2]: 9.104868  
weight[2][3][3]: 3.386664  
weight[2][3][4]: 6.302315  
weight[2][3][5]: -0.929242  
weight[2][3][6]: -3.404683  
weight[2][4][0]: -0.692689  
weight[2][4][1]: -3.877036  
weight[2][4][2]: 1.131659  
weight[2][4][3]: 4.158780  
weight[2][4][4]: -6.000619  
weight[2][4][5]: 4.651926  
weight[2][4][6]: 2.806753  
weight[2][5][0]: -5.806047  
weight[2][5][1]: 7.051200  
weight[2][5][2]: -2.145962  
weight[2][5][3]: -4.762430  
weight[2][5][4]: -2.382846  
weight[2][5][5]: 15.869929  
weight[2][5][6]: 6.559917  
weight[2][6][0]: -8.811863  
weight[2][6][1]: 3.983030  
weight[2][6][2]: 2.975153  
weight[2][6][3]: 2.605801  
weight[2][6][4]: 5.006668  
weight[2][6][5]: 4.914675  
weight[2][6][6]: 0.105588  
weight[3][0][0]: 6.295851  
weight[3][0][1]: -0.486709  
weight[3][0][2]: 3.503645  
weight[3][0][3]: -4.800145  
weight[3][0][4]: 2.609259  
weight[3][0][5]: 9.396222  
weight[3][0][6]: -8.576785  
network\_fuzzy\_number: 0.46000

[ VELOCITY DATA\_FILE : TAPPING, DATA\_GROUP : B ]

current error : 28.108394  
 SUCCESS in LEARNING WEIGHT at 3 iteration

```

num_layer: 4
num_node: 2 7 7 1
average: 15.500000 2.500000 8.786667
scale: 18.850000 1.950000 10.642667
threshold: 1.000000
beta: 1.000000
theta: 1.000000
weight[1][0][0]: 6.590302
weight[1][0][1]: -5.551017
weight[1][1][0]: 10.386615
weight[1][1][1]: -4.614928
weight[1][2][0]: 5.831252
weight[1][2][1]: 4.821404
weight[1][3][0]: -5.453854
weight[1][3][1]: -7.403156
weight[1][4][0]: -5.032909
weight[1][4][1]: -10.358948
weight[1][5][0]: 2.967225
weight[1][5][1]: 7.870755
weight[1][6][0]: 3.482381
weight[1][6][1]: -0.564650
weight[2][0][0]: -2.315311
weight[2][0][1]: 3.768299
weight[2][0][2]: 4.591960
weight[2][0][3]: -8.096088
weight[2][0][4]: 2.524873
weight[2][0][5]: -11.743852
weight[2][0][6]: 1.615078
weight[2][1][0]: -5.986081
weight[2][1][1]: 5.022068
weight[2][1][2]: -9.311971
weight[2][1][3]: -2.393853
weight[2][1][4]: 5.028524
weight[2][1][5]: 9.885856
weight[2][1][6]: -2.784843
weight[2][2][0]: -2.297184
weight[2][2][1]: 6.430166
weight[2][2][2]: -0.353231
weight[2][2][3]: 5.929295
weight[2][2][4]: -2.271768
weight[2][2][5]: 0.133771
weight[2][2][6]: 6.573627
weight[2][3][0]: -2.608638
weight[2][3][1]: 2.661118
weight[2][3][2]: 9.105088
weight[2][3][3]: 3.387504
weight[2][3][4]: 6.303095
weight[2][3][5]: -0.928482
weight[2][3][6]: -3.404363
weight[2][4][0]: -0.692409
weight[2][4][1]: -3.876196
weight[2][4][2]: 1.131919
weight[2][4][3]: 4.159100
weight[2][4][4]: -6.000339
weight[2][4][5]: 4.651706
weight[2][4][6]: 2.806773
weight[2][5][0]: -5.806507
weight[2][5][1]: 7.050200
weight[2][5][2]: -2.145062
weight[2][5][3]: -4.763190
weight[2][5][4]: -2.383686
weight[2][5][5]: 15.870249
weight[2][5][6]: 6.559857
weight[2][6][0]: -8.812023
weight[2][6][1]: 3.983510
weight[2][6][2]: 2.975533
weight[2][6][3]: 2.606581
weight[2][6][4]: 5.007328
weight[2][6][5]: 4.914995
weight[2][6][6]: 0.105408
weight[3][0][0]: 6.296651
weight[3][0][1]: -0.486149
weight[3][0][2]: 3.503945
weight[3][0][3]: -4.799565
weight[3][0][4]: 2.610059
weight[3][0][5]: 9.395882
weight[3][0][6]: -8.576725
network_fuzzy_number: 0.290000

```