GOVP1199812052

제1차 년도
최종보고서

건물공조시스템의 통합설계 S/W 개발

Development of the Integrated Computer Software for Building Air-conditioning System Design

건물공조시스템 설계용 S/W 개발

Development of the Computer Software for Building Air-conditioning System Design

연구기관

(주) 한 일 엠 이 씨
부설 한일기술연구소

과 학 기 술 부

제 출 문

과학기술부 장관 귀하

본 보고서를 "건물공조시스템의 통합설계 S/W 개발" 과제(세부과제 "건물공조시스템 설계용 S/W 개발")의 최종 보고서로 제출합니다.

1998. 8. 19

주관연구기관명 : 엔지니어링연구조합
총괄연구책임자 : 김 두 천
세부연구기관명 : (주) 한 일 엠 이 씨
부설 한일기술연구소
세부연구책임자 : 김 두 천
연 구 원 :

전 경 석(한일기술연구소)	류 해 성((주)장한기술)
여 명 석(〃)	방 규 원(〃)
이 용 희(〃)	양 재 구(〃)
하 승 범(〃)	이 두 호(〃)
최 철 훈(〃)	한 정 교((주)성아엔지니어링)
정 차 수((주)한일엠이씨)	김 주 석(〃)
정 경 일(〃)	임 상 준(서울대학교)
오 치 길(〃)	김 상 민(서울대학교)
국 찬 수(〃)	

요 약 문

I. 제 목

건물공조시스템 설계용 S/W 개발

II. 연구개발의 목적 및 필요성

건물 공조시스템의 성능은 공조시스템의 구성방식, 제어방식 및 시스템을 구성하는 열원기기나 공조기기의 운전효율에 의해 좌우된다. 따라서 시스템의 최 설계와 최적운전이 수반되어야만 최종 목적인 쾌적한 환경의 유지와 에너지 절약효과를 달성할 수 있다.

공조시스템의 최적설계 과정을 살펴보면 건물의 외피요소, 기상조건, 건물사용계획, 내부부하의 변동상황, 공조방식, 제어방식, 각종기기의 운전조건 및 성능, 실내온습도의 설정변동 등 여러가지 조건들을 가정하면서 시뮬레이션 계산을 행하고 검증을 거치면서 만족할 만한 결과에 이를 때까지 여러가지 조건들을 계속 수정하여 반복계산을 행하는 시행착오법에 의존하고 있기 때문에 엄청난 시간과 노력이 소요된다. 따라서 이러한 작업과정을 고급엔지니어의 인력에 의존하는 것은 경제적으로 불가능하기 때문에 선진국에서는 컴퓨터를 이용한 통합설계가 보편화되고 있다. 이러한 공조시스템의 통합설계 S/W는 설계과정에서의 노력과 시간이 절약될 뿐만 아니라 품질과 생산성을 제고시킬 수 있으며 최적설계화에 의한 에너지 절약효과도 매우 크다. 또한 설계과정에서 발생할 수 있는 오차와 오류를 감소시켜 기계설비시설물 등의 부실설계의 가능성을 최소화 할 수 있으며, 설계의 단순 반복과정을 전산화함으로서 설계의 경제성을 향상시키고 안정성 제고와 미관 및 환경친화적 시설물 확보에 기여할 수 있다.

III. 연구개발의 내용 및 범위

본 연구내용 및 범위는 전체연구기간 중 1차년도의 것으로 연구개발 수행내용 및 범위는 다음과 같다.

1. 자료조사 및 분석
   - 국내외 S/W개발 현황 조사
   - 실무분야의 의견조사 및 분석
   - 공조기 제조회사의 자료조사 및 분석
2. GUI 환경 설정
   - 프로그램의 사용환경 설정
   - 메뉴 구조설정
3. 기본 TOOL개발
   - 습공기 알고리즘 및 S/W 개발
   - 구성기기 알고리즘 개발
   - 유틸리티 알고리즘 개발
   - 공조방식 알고리즘 작성

IV. 연구개발결과

본 연구개발결과는 전체연구기간중 1차년도의 연구에 의한 것으로 다음과 같다.

1. 자료조사 및 분석

   주요조사 실적은 다음과 같다.
   - DOE-2, HASP, E-20II
   - Elite사의 S/W Set
   - HVAC2 TOOLKIT
2. GUI 환경 설정

   주요 GUI 환경설정 항목은 다음과 같다.

- 하드웨어 : 586 이상
- 언어 : C/C++
- 주메뉴 8종

3. 기본 TOOL개발

- 습공기선도 해석 S/W 개발 (PSYALL)
- 습공기과정 해석 S/W 개발
- 구성기기 알고리즘 작성(팬, 펌프, 밸브)
- 유틸리티 알고리즘 작성(응답계수, 전달함수 등)
- 공조시스템 알고리즘 16종 작성

V. 연구개발결과의 활용계획

본 연구결과는 전체연구기간중 1차년도의 연구결과로 완전한 연구결과가 아님으로 실무적으로 활용하기에는 보다 많은 부분이 보완되어야 할 것이다. 다만, 본 연구결과로 얻어진 습공기선도 해석 및 과정해석 S/W는 건축설비설계사무실, 건설회사와 관련 대학 등에 배포하여 실무 및 교육용으로 활용하도록 하겠으며, 연구결과 작성된 알고리즘은 향후 프로그램으로 개발하여 실무에 활용할 수 있도록 하겠다.

S U M M A R Y

I. Title

Development of the Computer Software for Building Air-conditioning System Design

II. Objectives and Necessity of the Research

The performance of the building air-conditioning system is depend on the types of HVAC system, control system and operating efficiency of the system components.

Therefore, optimum design and optimum operation is necessary for the and energy saving and comfort.

Optimum design of HVAC systems is usually being done by iteration procedures which follows, on assumption of the buildings envelope performances, climate conditions, building operations, internal load changes, air-conditioning system types, control types, working conditions and performance of the components, room temperature & humidity settings etc, calculation and examination of the results in every sequence such as room loads, component loads, plant loads, energy consumptions until the designer could be satisfied. As much time and effort is required for the senior level engineers to do this procedure, an effective engineering tool based on Integrated computer software for the optimum Building HVAC System design, which is already widely used in developed countries, should be provided urgently for Korean Engineering Industry.

This computer software enables to be economical design, economical operation, fine sight and environmentally compatible buildings.

III. Contents and Scope of the Research

The contents of the first year research is as follows.

1. Review of the technical data
2. Configuration of the graphical user interface(GUI) environment
3. Development of the basic design tool

IV. Results of the Research

The results of the research is described at each chapter and summarized as follows.

1. Review of the technical data
   - DOE-2, HASP, E-20 II
   - Elite s/w co. (S/W Set)
   - HVAC2 TOOLKIT

2. Configuration of the GUI environment
   - Hardware : above 586
   - Language : C/C++
   - Main menu : 8 each

3. Development of the Basic Design Tool
   - Development of the software for the psychometric chart analysis (PSYALL)
   - Development of the software for the moist air process analysis

- Development of the algorithm for the HVAC components (Fan, Pump, Valve)
- Development of the algorithms for the utilities(heat transfer, shadow, regression)
- Development of the algorithms for the secondary air-conditioning system

V. Proposals for Utilization of the Results

Because of the first year research, the results is needed to be updated in the succeeding year for utilization.
But the softwares for the psychometric chart and moist air process analysis in this research can be profitably used in design office, construction company, university.

C O N T E N T S

Chapter 1. Introduction ··························· 1
Section 1. Necessity ··························· 1
Section 2. Objectives and contents of the research ··········· 2

Chapter 2. State of the arts ························· 4
Section 1. State of the arts ························ 4
Section 2. A problem of the domestic situation ············· 5

Chapter 3. Contents and results of the research ············· 7
Section 1. Configuration of the GUI environment ············· 7
Section 2. Development of the algorithm and software for psychometric chart analysis ························ 10
Section 3. Development of the algorithm for HVAC components ····· 32
Section 4. Development of the algorithm for utilities ········ 45
Section 5. Development of the algorithm for air-conditioning system · 76

Chapter 4. Accomplishments and contributions of development ······ 147
Section 1. Accomplishments ························ 147
Section 2. Contributions of the research ················ 147

Chapter 5. Proposal for utilization ···················· 148

Chapter 6. References ···························· 149

Appendix A. Program for Psychometric analysis ·········· 153
B. Program for HVAC components analysis ········ 183
C. Program for transient heat transfer analysis ··· 191
D. Program for shadow chart analysis ·········· 206
E. Program for regression analysis ··········· 211

목 차

제 1 장 서 론 1
제 1 절 연구개발의 필요성 1
제 2 절 연구개발의 목표 및 내용 2

제 2 장 국내외 기술개발 현황 4
제 1 절 국내외 연구현황 4
제 2 절 국내기술 현황의 취약성 5

제 3 장 연구개발수행 내용 및 결과 7
제 1 절 GUI 환경설정 7
제 2 절 습공기 알고리즘 작성 및 S/W 개발 10
제 3 절 구성기기 알고리즘 작성 32
제 4 절 유틸리티 알고리즘 작성 및 S/W 개발 45
제 5 절 공조방식 알고리즘 작성 76

제 4 장 연구개발목표 달성도 및 대외기여도 147
제 1 절 연구개발 목표 및 달성도 147
제 2 절 연구개발의 대외 기여도 147

제 5 장 연구개발결과의 활용계획 148

제 6 장 참고문헌 149

부록 A. 습공기상태량 계산 프로그램 153
B. 구성기기 계산 프로그램 183
C. 열전달 계산 프로그램 191
D. 그림자 계산 프로그램 206
E. 회귀분석 프로그램 211

제 1 장 서 론

제 1 절 연구개발의 필요성

건물 공조시스템의 성능은 공조시스템의 구성방식, 제어방식 및 시스템을 구성하는 열원기기나 공조기기의 운전효율에 좌우되므로 시스템의 최적설계와 최적운전이 수반되어야만 최종 목적인 쾌적한 환경의 유지와 에너지 절약효과를 달성할 수 있다.

공조시스템의 최적설계 과정은 건물의 외피요소, 기상조건, 건물사용계획, 내부부하의 변동상황, 공조방식, 제어방식, 각종 기기의 운전조건 및 성능, 실내온습도의 설정변동 등 여러가지 조건들을 가정하면서 시뮬레이션 계산을 행하고 검증을 거치면서 만족할 만한 결과에 이를 때까지 여러가지 조건들을 계속 수정하여 반복계산을 행하는 시행착오법에 의존하고 있기 때문에 엄청난 시간과 노력이 소요된다. 따라서 이러한 작업과정을 고급엔지니어의 인력에 의존하는 것은 경제적으로 불가능하기 때문에 선진국에서는 컴퓨터를 이용한 통합설계가 보편화되고 있다.
다음의 그림은 통합설계와 관련된 S/W의 현황이다.

공기조화·냉동공학회가 장비선정을 위한 최대부하계산프로그램인 LOADCALC를 개발완료, 보급하게 되었고, 에너지계산용 시뮬레이션프로그램도 개발을 시작하였기 때문에 기상조건, 건물의 사용실태, 건축재료의 열특성, 기기·장비의 효율, 공조설비방식, 운영방식관리, 에너지 가격체계 등등 여러 가지 상이한 조건 때문에 외국의 S/W 사용시에 겪게되는 애로사항을 해소할 수 있게 되었다. 그러나 현장의 설계실무자에게 필요한 설계계산용 Toolkit은 전혀 없는 실정이다. 이러한 설계 계산용 Toolkit은 설계과정에서 발생할 수 있는 오차와 오류를 감소시켜 기계설비시설물 등의 부실 설계의 가능성을 최소화 할 수 있으며, 설계의 단순 반복과정을 전산화함으로서 설계의 경제성을 향상시키고 안정성 제고와 미관 및 환경친화적 시설물의 확보에 기여할 수 있다. 따라서 Elite Software 사의 S/W 셀과 ASHRAE의 HVAC2 Toolkit과 같은 설계용 S/W의 개발이 시급히 요구되고 있다.

공조설비 설계순서	설 계 도 구	비 고
계 획	* 간이부하계산	
	* 소요 SPACE 산정	
↓		
시스템 선정	* 년간부하계산	공조학회개발중
	* 시스템시뮬레이션	
	* 경제성 평가	
↓		
설계계산 및 도면작성	* 상세부하계산	LOADCALC (공조학회개발)
	* 장비선정	
	* 배관계산	
	* 덕트계산	
	* 자동작도 - 배관 - 덕트 - 장비 - 시공상세	

통합설계 관련 S/W 현황

제 2 절 연구개발의 목표 및 내용

1. 연구개발의 최종목표

건물공조시스템의 통합설계를 위한 국내실정에 부합되는 설계용 S/W의 개발

2. 1차년도 연구개발 목표 및 내용

가. 자료조사 및 분석

- 국내외 S/W 개발 현황 조사
- 실무분야의 의견조사 및 분석
- 공조기 제조회사의 자료조사 및 분석

나. GUI 환경 설정

- 프로그램의 사용환경 설정
- 메뉴구조 설정

다. 기본 TOOL 개발

- 습공기 알고리즘 및 S/W 개발
- 구성기기 알고리즘 개발
- 유틸리티 알고리즘 개발
- 공조방식 알고리즘 작성

제 3 절 연구개발 추진체계

1. 추진전략 및 방법

국내의 설비설계회사의 실무현황과 의견 수렴을 통한 실무용 통합설계 S/W의 기능을 설정하고, 이와 연계 가능한 기본자료를 수집하여 우리 현실에 맞는 설계용 S/W를 구성한다. 국내에 사용중인 공조방식의 종류와 공조기기의 성능에 대해서는 국내 제조업체의 관련자료와 외국의 데이터를 비교, 분석하여 적용한다.

설계기준의 작성은 국내외 각 설계사무소의 현황을 파악, 검증한 후에 기준을 제시한다.

본 연구는 공조시스템의 통합설계 S/W 개발로 세부과제간 긴밀한 상호협의와 검증을 거쳐 연구를 진행한다.

2. 추진체계

1차년도의 연구개발 추진체계는 다음과 같다.

자료조사 및 업무분석

GUI 환경설정

2차공조시스템 알고리즘 개발

HVAC Toolkit 알고리즘 개발

상 호 검 증 및 협 의

제 2 장 국내외 기술개발 현황

제 1 절 국내외 연구현황

1. 외국의 경우

동적 열부하 계산법의 대표적인 프로그램으로 일본에서는 응답계수법을 이용한 HASP/ACLD-8001, 미국에서는 응답계수법에 전달함수법을 가미한 에너지성의 DOE-2 등의 전산프로그램 등이 이용되고 있으며, 상용으로 시판되고 있는 전산 프로그램 패키지는 300여개에 이르고 있다. 일본의 HASP/ACLD-8001은 BASIC으로 된 MICRO/HASP/1982로도 개발되었으며, 최근에는 그 일부를 수정 보완하여 열부하계산 프로그램인 HASP/ACLD-8501와 공조시스템의 표준 시뮬레이션 프로그램인 HASP/ACSS-8502를 개발하여 보급하고 있다.

한편 미국에서는 지금까지 사용해 오던 DOE-2.1C(1984)를 수정하여 DOE-2.1E를 보급하고 있으며, DOE-2.1D에 의한 마이콤용의 MICRO-DOE2, ADM-DOE2 등이 보급되어 있다. 대표적인 설계계산용 프로그램은 다음과 같다.

외국의 대표적인 설계계산용 프로그램

프로그램명	개 발 자	용 도	평 가
DOE-2 TRNSYS TRACE HASP	미국 NBS 미국 Wisconsin Madison Univ. 미국 TRANE Co. 일본 공기조화·위생공학회	연구용(정밀계산용) 연구용(기기, 시스템 시뮬레이션) 상 업 용 동적열부하계산	대형 컴퓨터용으로 매시간별 동적계산 방법이며, 사용이 매우 복잡하여 주로 연구용으로 이용된다. 전문가가 필요하다.
BESA BUILDSYM TrakLoad	(캐나다)CANDAPLAN RESOURCES, INC. (스웨덴)FLAKT AB (미국)MORGAN	개보수, 진단용 상 업 용 상 업 용(진단용)	PC용 프로그램이나 우리실정에 맞지 않고, SOURCE CODE를 알 수 없으며, 신뢰도를 확인하기가 어렵다.

한편 실무설계용으로는 Elite Software 사의 습공기계산용 PSYCHART 등 22개로된 PC용 S/W 등이 있으며, ASHRAE의 HVAC2 Toolkit 등이 보급되고 있다.

2. 국내의 경우

장비선정을 위한 최대부하계산용으로 일본 공기조화·위생공학회의 실효온도차법에 의한 실내부하계산용인 윈도우 풀다운 메뉴시스템의 PC용 한글 S/W인 PEAKLOAD와 설계실무에 사용되고 있는 장한기술산업의 LOADSYS가 있으며, 공기조화·냉동공학회에서는 1996년 9월에 통상산업부의 지원하에 설계실무용 프로그램인 LOADCALC를 개발하여 보급을 준비하고 있다.

실무용 Toolkit으로는 디씨에스사의 CO-ME가 있으나, 이것은 도면작성에 국한되어 있다.

제 2 절 국내기술 현황의 취약성

국내에서도 공기조화·냉동공학회가 장비선정을 위한 최대부하계산프로그램인 LOADCALC를 개발완료, 보급하게 되었고, 에너지계산용 시뮬레이션프로그램도 개발을 시작하였기 때문에 기상조건, 건물의 사용실태, 건축재료의 열특성, 기기·장비의 효율, 공조설비방식, 운영방식관리, 에너지 가격체계 등등 여러 가지 상이한 조건 때문에 외국의 S/W 사용시에 겪게되는 애로사항을 어느정도 해소할 수 있으리라 본다. 그러나 현장의 설계실무자에게 필요한 설계계산용 Toolkit은 전혀 없다. 따라서 Elite Software 사의 S/W 셀과 ASHRAE HVAC2 Toolkit과 같은 설계용 S/W의 개발이 시급히 요구되고 있으며, 설계업무의 체계화 및 표준화를 위한 설계의 계획과 실행과정의 통합화가 필요하다.

제 3 장 연구개발수행 내용 및 결과

제 1 절 GUI 환경설정

1. 하드웨어의 요구사항

  가. 시스템 CPU : 586급 이상

  나. RAM : 8M 이상

  다. HARD DISK 여유 공간 : 50M byte 이상

2, 사용 O/S

  WINDOWS 95 / NT 이상

3. 프로그램 사용언어

  C / C++

4. 프로그램의 사용의 특징

  가. 마우스의 이용

자료 입력시 마우스를 사용하여, 커서를 원하는 위치에 빠르게 이동시켜 주어서, 키보드의 조작을 최소화 시켜 줄 수 있다.

  나. 풀다운 메뉴(PULL-DOWN MENU)방식

프로그램 사용자가 하위 메뉴에 포함된 항목을 보고 싶을 때, "Alt"나 "F10" 키를 사용하면 화면에 풀다운 메뉴가 나타나며, 여기서 키보드의 방향키을 사용하여 원하는 항목을 손쉽게 선택 할 수 있다.

주메뉴에 있는 각 항목을 마우스로 선택하면 하위 메뉴가 나타나면서 계속 원하는 항목을 선택할 수 있는데 이러한 진행 방식을 "풀다운 메뉴 방식"이라 한다.

  다. 아이콘 툴바

사용자가 자주 사용하는 기능을 보다 빠르게 실행시키기 위하여 풀다운 메뉴 아래에 아이콘 툴바를 두었다. 따라서 사용자는 그림 모양의 아이콘 툴바를 마우스로 클릭함으로서 빠르고 편리하게 프로그램을 사용할 수 있다.

  라. 트리(TREE)방식의 메뉴구조 윈도우

화면의 왼쪽에 트리방식으로 메뉴를 보여줄 수 있으며 여기서 사용자가 원하는 프로그램의 항목을 마우스로 클릭함으로서 풀다운 메뉴 방식보다 빠르게 프로그램을 실행 시킬 수 있다.

마. 하이퍼 텍스트 방식의 도움말 기능

사용자가 프로그램을 사용하다가 <F1>을 누르거나 아이콘 툴바의 도움말을 마우스로 누르면 자세한 설명을 볼 수 있다. 도움말 화면에서 사용자가 원하는 항목을 마우스로 클릭하여 이동하면서 설명을 읽을 수 있으며 찾아보기에서 원하는 키워드를 입력하여 관련 설명을 볼 수도 있다. 도움말 내용중의 파란색 글자로 표시된 부분은 그 항목과 관련이 있는 다른 도움말과 연결되도록 하이퍼 텍스트 방식으로 되어 있는 것으로 이 부분을 마우스로 클릭하면 해당 항목의 도움말로 이동하게 된다.

5. 전체메뉴의 구성

전체메뉴의 구성은 다음과 같다.

주메뉴	1차 서브메뉴	2차 서브메뉴	3차 서브메뉴
1. 습공기계산	상태량계산		
	과정해석	단순과정(4)	가열
			냉각
			가습
			혼합
		복합과정	가열가습
			혼합냉각
			혼합냉각재열
			혼합냉각바이패스
	시스템해석	(16개)	
2. 열원기기 (1차공조시스템)	냉동기		
	보일러		
	열병합시스템	디젤엔진	
		가스터빈	
	축열조		
	냉각탑		
	펌프		
	열교환기		
	제어방식	바이패스제어	
		대수제어	
		열회수운전	
		병렬운전	
	특성데이터		

주메뉴	1차 서브메뉴	2차 서브메뉴	3차 서브메뉴
3. 공조방식 (2차공조시스템)	정풍량방식(CAV)	단일구역방식	
		단말재열방식	
		이중덕트방식	
		이중덕트 Draw-Thru 냉각방식	
		팬코일방식	
		멀티존방식	
	변풍량방식(VAV)	외주부방열기방식	
		외주부공기난방방식	
		단말재열장치	
		병렬팬급기방식	
		직렬팬급기방식	
	패키지에어콘		
	히트펌프	공기열원	
		수열원	
	팬코일유닛방식		
	환기조화기방식		
	외기조화기방식		
4. 구성기기	단순유동기기	팬	
		펌프	
		밸브	
	열전달장치	열교환기(3종)	
		코일(12종)	
	단말기기 (터미널유닛)	변풍량유닛	
		정풍량유닛	
		변풍량팬급기유닛	
		급기디퓨져(CAV방식)	
	제어장치		
5. 유틸리티	일반	단위변환	
		수치해석 Tool	
	열전달기초	복사형상계수	
		대류열전달계수	
		가중계수계산	
	일사량	ASHRAE 일사량모델	
		직산분리모델	
		건물그림자계산	
	부하계산	부하구성요소	
		벽의 열취득	열관류율
			응답계수
			전달함수
		창의 열취득	차폐계수
			그림자면적
			일사열취득
6. 최대부하계산			
7. 경제성평가			
8. 데이터베이스			

제 2 절 습공기 알고리즘 작성 및 S/W 개발

1. 개요

습공기의 상태량 계산에는 Keenan의 식, ASME 공식 및 Goff-Gratch[1] 공식 등을 사용해 왔으나 국제실용 온도눈금과 SI단위계의 적용으로 인하여 1981년 이후에는 ASHRAE에서 개발한 Wexler-Hyland[2,3]의 공식이 사용되고 있다.

그러나 이 방법은 실제기체로서의 상태량을 계산하기 때문에 알고리즘이 매우 복잡하여, ASHRAE는 2~3자리의 유효숫자로 만족할 수 있는 공기조화계산에는 습공기를 이상기체로 간주한 근사계산법을 권장하고 있다[3]. 이것은 기체의 포화수증기압이 물에 용해되어 있는 공기, 압력변화에 따른 물의 상태량 변화 및 습공기 분자간의 상호작용 때문에 Wexler-Hyland[4,5] 식으로부터 계산한 순수한 물의 포화수증기압과는 약간의 차이가 생기는데 이것을 수정계수(enhancement factor)로 보정하는 방법이며, 압력만을 고려한 Wexler-Hyland의 식과 압력과 온도를 모두 고려한 手塚後一[6,7]의 식 등이 있다.

습공기 상태량 계산 프로그램으로는 ASHRAE의 HVAC2 TOOLKIT, 일본공기조화・위생공학회 및 Elite 사의 Psychen 등이 있으며, 국내에는 공기조화・냉동공학회의 습공기선도 작성용 프로그램이 있다.

본 연구에서는 공기조화시스템 설계에서 가장 기본이 되는 습공기의 열역학적 상태량계산방법을 검토하고, 습공기의 과정해석과 습공기선도를 작도하기 위한 PC용 소프트웨어를 개발하였다. 습공기 상태량 계산 알고리즘은 HVAC2 TOOLKIT를 골간으로 하고 엔탈피의 정밀도를 향상시키기 위하여 手塚後一의 수정계수의 식을 -40℃ ~ 80℃의 온도범위로 확장하여 사용하였으며, 실용자의 편의성을 위하여 마이크로소프트 윈도우 95와 NT의 GUI 환경에서 사용할 수 있도록 하였다.

2. 계산 알고리즘

가. 습공기의 이상기체 관계식

습공기의 상태량은 습공기를 건공기와 수증기의 혼합기체로 생각하고, 그것들을 모두 이상기체로 취급하면 간단하게 계산할수 있다.

절대습도는 습공기 혼합물에서 건공기의 질량에 대한 수증기 질량의 비인데, 전압력 P와 수증기분압 PW로 표시하면 다음과 같다.

$$W = 0.62198\ P_w/(P - P_w) \tag{2.1}$$

상대습도는 동일한 온도와 압력하에서 포화습공기에 대한 임의상태에서의 수증기의 몰비의 비이다.

$$\Phi = \left.\frac{x_w}{x_{ws}}\right|_{t,\,p} = \left.\frac{P_w}{P_{ws}}\right|_{t,\,p} \tag{2.2}$$

비체적은 건공기 1kg에 대한 습공기의 체적이므로 상태방정식으로부터 다음과 같이 표시된다.

$$v = \frac{0.2871\,T}{P}(1 + 1.6078\,W) \tag{2.3}$$

이상기체 혼합물의 엔탈피는 각 성분의 엔탈피의 합과 같다.

$$h = 1.006\,t + W(2501 + 1.805\,t) \tag{2.4}$$

나. 습구온도 및 노점온도

열역학적 습구온도 t^*는 공기(t, W, h)가 정압과정에서 다량의 수분과 접하며 긴 단열유로를 통과하면서 온도 t^*의 물로부터 수분을 공급받아서 물과 같은 온도의 포화공기(t^*, W_s^*, h_s^*)가 되어 나오는 과정을 생각했을 때의 온도이며, 다음식을 이용한 반복계산으로 구한다.

$$W = \frac{(2501 + 1.805\,t^* - h_w^*)W_s^* - 1.006(t - t^*)}{2501 + 1.805\,t - h_w^*} \qquad (2.5)$$

여기서, h_w^*는 습구온도 t^*에서의 물의 비엔탈피이며 다음의 근사식으로 구한다.

$$h_w^* = -333.43 + 2.031\,t^*, \quad t^* \le 0 \qquad (2.6)$$

$$h_w^* = 4.186\,t^*, \quad t^* > 0 \qquad (2.7)$$

노점온도 t_d는 ASHRAE편람 등에서는 대부분 경험식을 사용하고 있으나 오차를 최소화하기 위하여 다음 식을 이용한 반복계산으로 구한다.

$$W_s(P, t_d) = W \qquad (2.8)$$

포화수증기압 P_{ws}는 Wexler-Hyland의 식 (2.3)으로부터 구한다.

다. 수정계수(enhancement factor)

습공기를 이상기체로 가정함으로서 발생되는 오차를 줄이기 위하여 ASHRAE 등은 수정계수를 적용하여 절대습도를 보정하여 계산의 정밀도를 높이고 있는데, 그 적용범위는 0℃ ~ 50℃ 및 압력 50기압 이하이다. 본 연구에서는 적용온도범위를 -40℃ ~ 80℃로 확장하기 위하여 온도변화에 따른 보정을 추가한 다음 식을 적용하였다.

$$P_{ws} = f_s\,P_{ws} \qquad (2.9)$$

여기서, f_s는 수정계수이며 다음과 같다.

$$f_s = 1 + 0.004P_r + c_1 + c_2 t + c_3 t^2 + c_4 t^3 + c_5 t^4 + c_6 t^5 + c_7 t^6 \qquad (2.10)$$

여기서,

c_1 = 0.162867E-04, c_2 = -0.687144E-05

c_3 = 0.632170E-06, c_4 = 0.111286E-08

c_5 = 0.178558E-10, c_6 = -0.471039E-12

c_7 = -0.773271E-14

이고, P_r는 표준대기압에 대한 국지대기압의 비이다.

라. 계산결과의 검토

표 2.1과 표 2.2는 본 연구에서 채택한 계산알고리즘과 참고문헌 2,7,8)에 의해 계산한 포화상태의 엔탈피와 비체적을 비교한 것이다.

온도범위 -40℃ ~ 80℃에서 엔탈피는 0.005%, 비체적은 0.1%의 오차범위로서 충분한 정밀도를 보여주고 있다.

표 2.1 포화상태에서의 엔탈피 비교(kJ/kg)

P = 101.325 kPa

Temperature	ASHRAE RP-216	**PSYALL**	SHASE	HVAC2 TOOLKIT
-40	-40.04	**-40.05**	-39.81	-40.05
-20	-18.55	**-18.55**	-18.44	-18.56
0	9.47	**9.48**	9.43	9.44
20	57.55	**57.56**	57.26	57.41
40	166.67	**166.68**	165.68	166.03
60	460.85	**460.87**	457.26	458.08
80	1541.66	**1541.75**	1524.33	1527.43

표 2.2 포화상태에서의 비체적 비교(㎥/kg)

P = 101.325 kPa

Temperature	ASHRAE RP-216	**PSYALL**	SHASE	HVAC2 TOOLKIT
-40	0.6598	**0.6606**	0.6605	0.6606
-20	0.7172	**0.7179**	0.7178	0.7179
0	0.7781	**0.7785**	0.7784	0.7785
20	0.8498	**0.8501**	0.8500	0.8501
40	0.9568	**0.9569**	0.9566	0.9569
60	1.1752	**1.1751**	1.1746	1.1751
80	1.8810	**1.8803**	1.8784	1.8803

3. 프로그램의 개발

가. 기본구성

본 연구에서는 습공기의 상태량 계산방법을 검토하고 이를 토대로 PC용 습공기 상태량 계산 및 과정해석용 소프트웨어인 PSYALL을 개발하였다. PSYALL을 활용하는 실무진에서의 사용의 편의성을 위해 마이크로소프트 윈도우95와 NT의 GUI 환경에서 사용할 수 있도록 하였고, 과정해석은 단일과정해석과 복합과정해석을 선도상에 비주얼하게 표시할 수 있도록 구성하였다. 또한 향후 프로그램의 유지, 보수 및 확장성과 계산의 정확성을 고려해 C/C++를 이용해 코드를 작성하였고, 전체적으로 알고리즘의 정확도와 계산시간 등을 고려해 코드를 구성하였다. 습공기선도의 작도는 화면상의 윈도우 크기에 대한 좌표 및 옵션설정을 통해 여백을 제외한 실제 작도를 위한 영역을 설정하고, 가변적인 윈도우의 크기에 대해 확대/축소가 가능한 선도를 구성하였다.

건구온도와 절대습도를 기본 X, Y축으로 하여 각각의 상태량의 변화에 따른 내부 좌표값을 계산하여 컴퓨터의 화면상에 실시간 작도를 한다. 모든 상태량의 작도에서 건구온도, 절대습도 그리고 상대습도의 범위 등의 한계치에서 각 상태량의 좌표값이 벗어나는 것을 방지하기 위해 2차원 클리핑을 통해 선도안의 내부에서만의 각 상태량을 계산하여 작도한다. 그림 2.1은 습공기선도 작도를 위한 기본정보의 구조

체이며, 그림 2.2는 함수의 흐름도이다.

```
typedef struct TFrame
{
      int     RealHeight;      // 그래프 영역의 크기
      int     RealWidth;
      int     Height;          // 그래프 내부 영역의 크기
      int     Width;
      int     x1,y1,x2,y2;     // 그래프 내부 영역의 좌표
      int     RulerXn;         // 눈금의 수
      int     RulerYn;
      int     RulerXv;         // 눈금의 크기
      int     RulerYv;
      float   XSkip;
      float   YSkip;
      float   XRatio;          // 화면비율
      float   YRatio;
      int     RulerSx,RulerEx;    // X 눈금
      int     RulerSy,RulerEy;    // Y 눈금
      struct TGap Gap;            // 그래프박스안의 여백
}TFrame;
```

그림 2.1 TFrame 구조

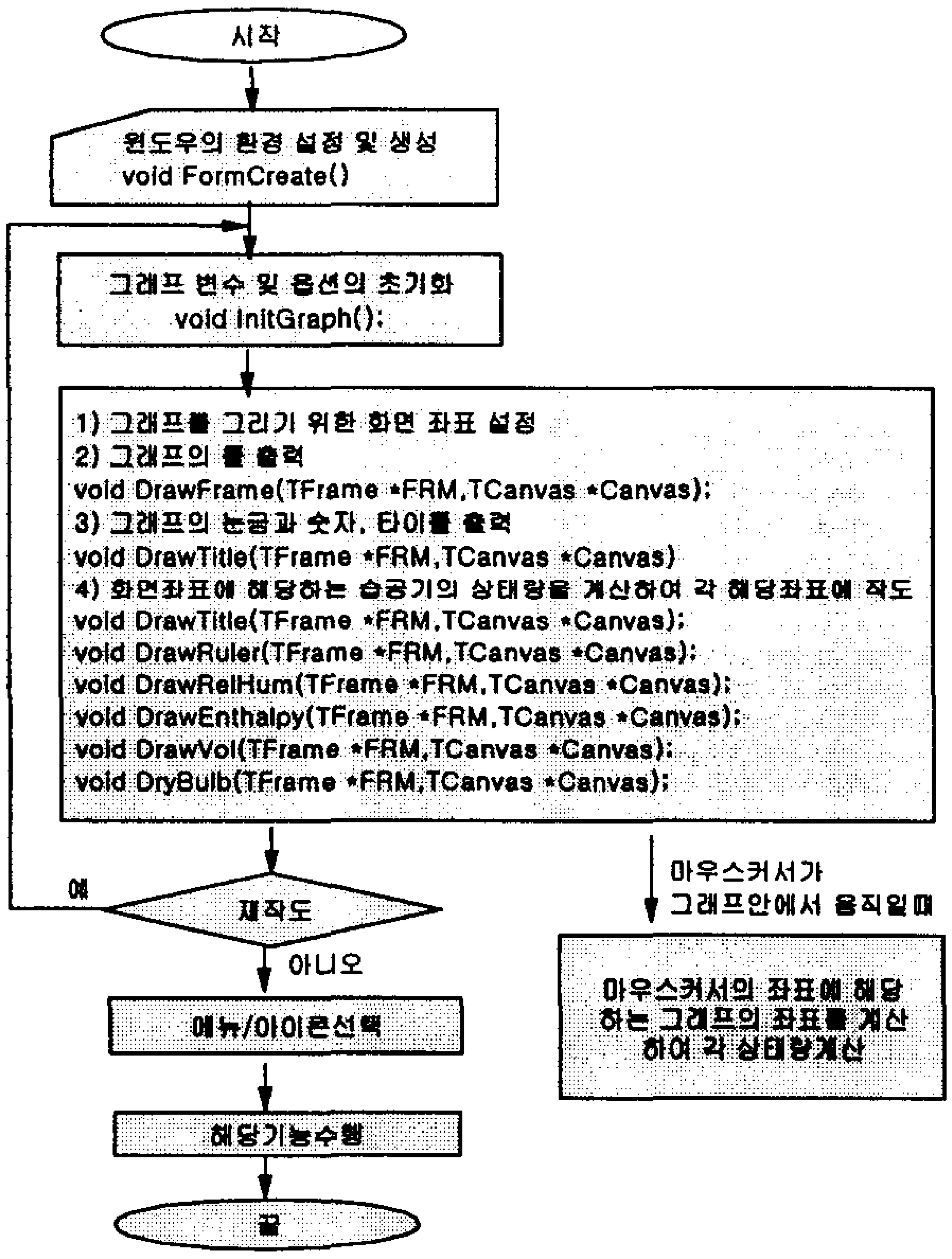


그림 2.2 함수의 흐름도

나. 프로그램의 사용

(1) 상태량 계산

(가) 프로그램의 구성

습공기의 상태는 세가지의 상태량을 알아야 구할 수 있는데, 그 대표적인 3가지 상태량은 압력, 건구온도 및 습도에 관한 어떤 하나의 상태량이며, 습도를 나타내는 상태량으로는 절대습도, 상태습도, 노점온도, 습구온도 등이 있다. 따라서 압력이 주어지면 건구온도와 습도관련 상태량의 조합에 의해 15종의 입력방식이 있으나, 본 프로그램은 표 2.3과 같은 실용적인 10종에 관한 서브루틴으로 구성하였다.

표 2.3 PSYALL의 Subroutine

Subroutine명	입력변수	Subroutine명	입력변수
TDB_RH	압력, 건구온도, 상대습도	RH_W	압력, 상대습도, 절대습도
TDB_TWB	압력, 건구온도, 습구온도	W_H	압력, 절대습도, 엔탈피
TDB_W	압력, 건구온도, 절대습도	W_WTB	압력, 절대습도, 습구온도
TDB_H	압력, 건구온도, 엔탈피	RH_H	압력, 상대습도, 엔탈피
TDB_TDP	압력, 건구온도, 노점온도	RH_TWB	압력, 상대습도, 습구온도

(나) 계산순서

1. 상태량계산을 위한 10가지의 계산루틴 중 한개를 선택한다.
2. 표고선택을 통해 압력을 계산한다.
3. 두가지의 상태량값을 입력한다.
4. 두가지의 상태량값 입력시 '범위계산' 버튼을 통해 올바른 입력범위를 설정한다.
5. '계산' 버튼을 통해 상태량을 계산한다.

(다) 계산 예제

상태량계산 예제로서 그림 2.3은 서울지역(고도 85.5m)에서 건구온도 10℃, 상대습도 50%를 입력하여 절대습도 3.846 g/kg, 엔탈피 19.75 kJ/kg, 습구온도 5.50℃, 노점온도 0.06℃, 비체적 0.815 m^3/kg의 상태량을 계산한 예이다.

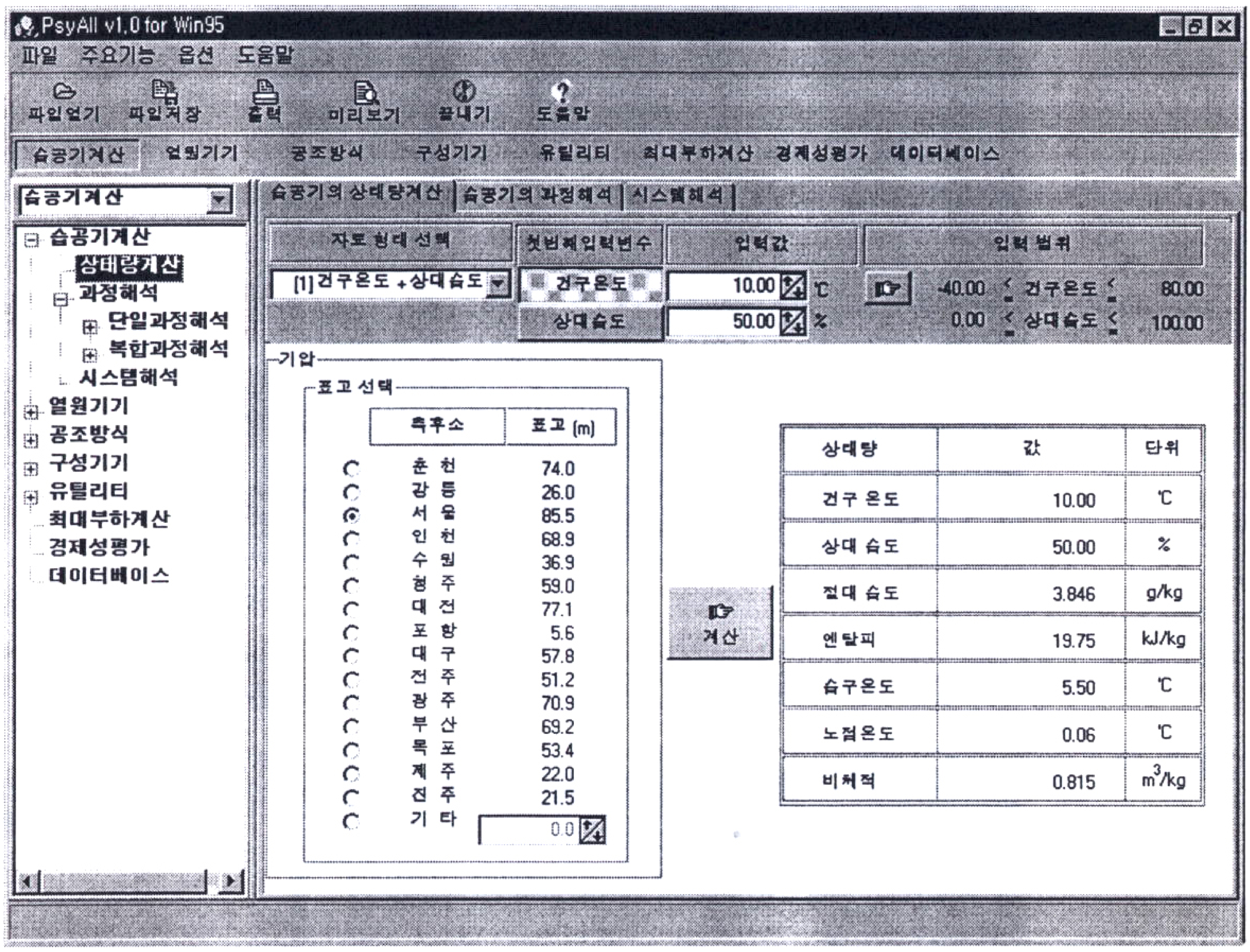


그림 2.3 습공기 상태량 계산 예제 화면

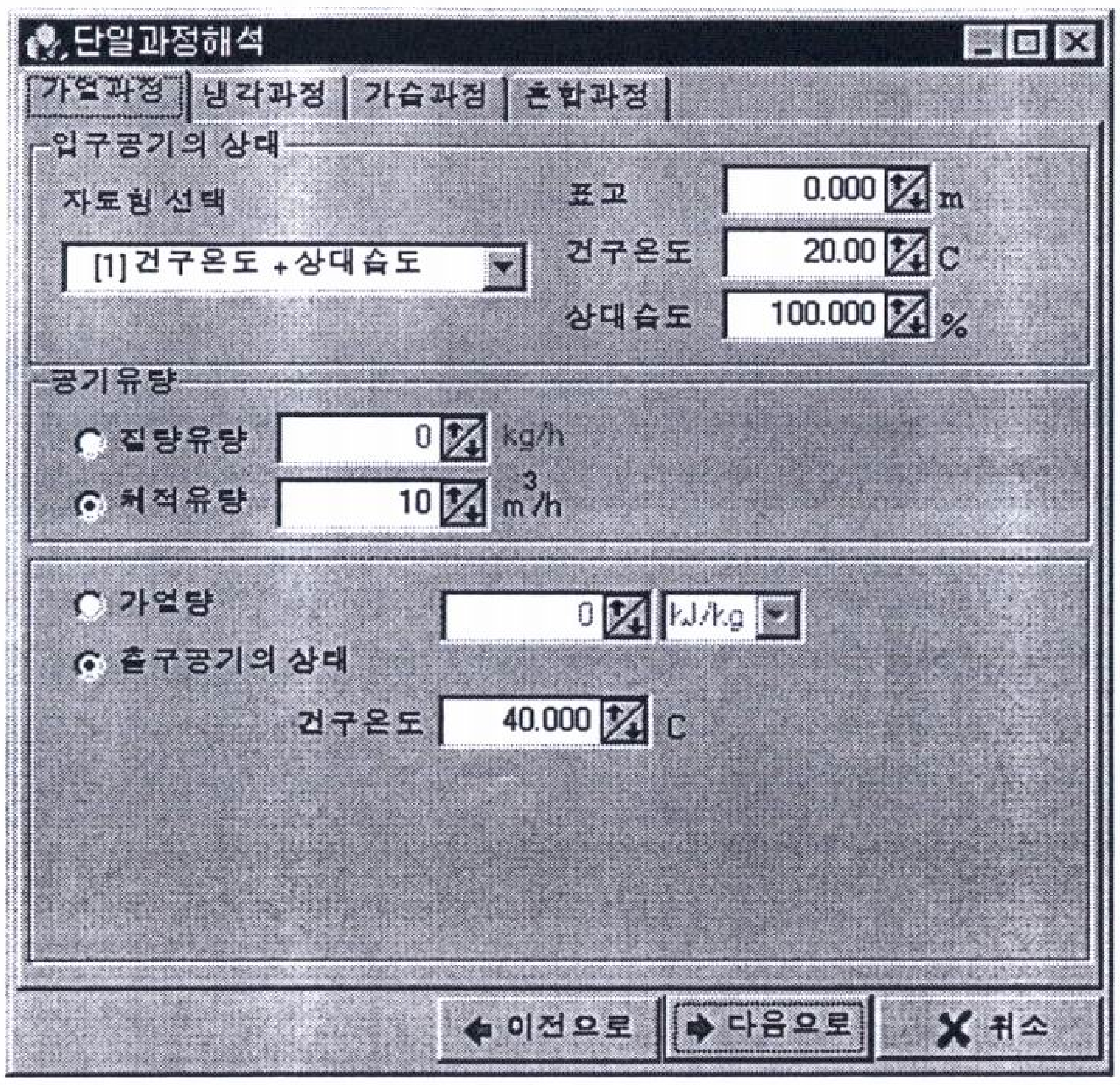
(2) 과정해석

PSYALL에서의 과정해석은 가열, 냉각, 가습, 혼합과정의 단일과정과 가열가습, 혼합냉각, 혼합냉각 및 재열, 혼합냉각 및 바이패스의 복합과정을 취급하고 있다.

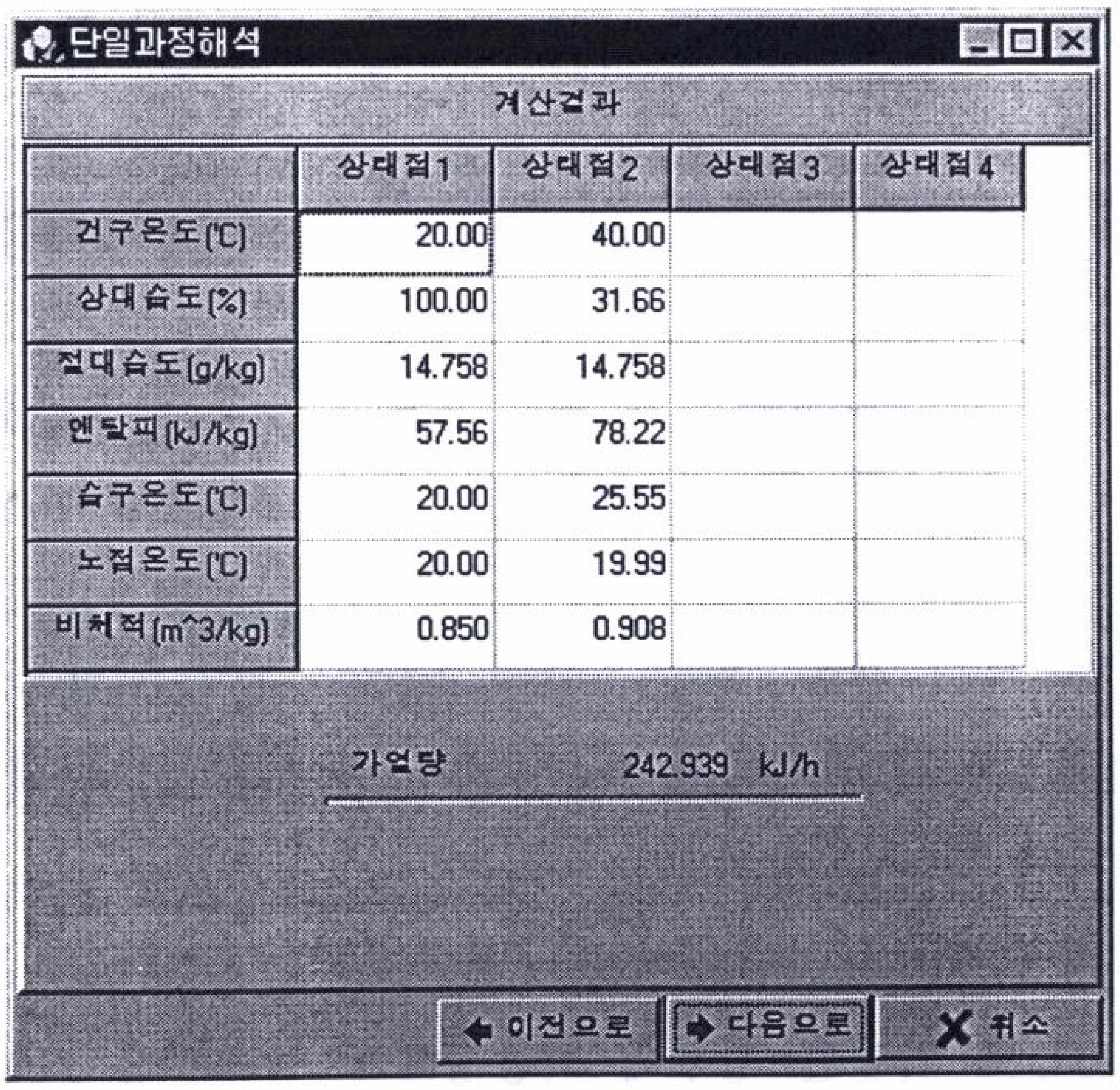
(가) 가열과정 해석

사용 예로서 표고가 0.0 m인 지역에서 건구온도 20℃, 상대습도 100%인 공기 10 m^3/h를 건구온도 40℃로 가열한 경우에 최종상태의 상태량과 가열량을 계산하고 이과정을 습공기선도에 작도하면 다음과 같다.

1. 입력 화면

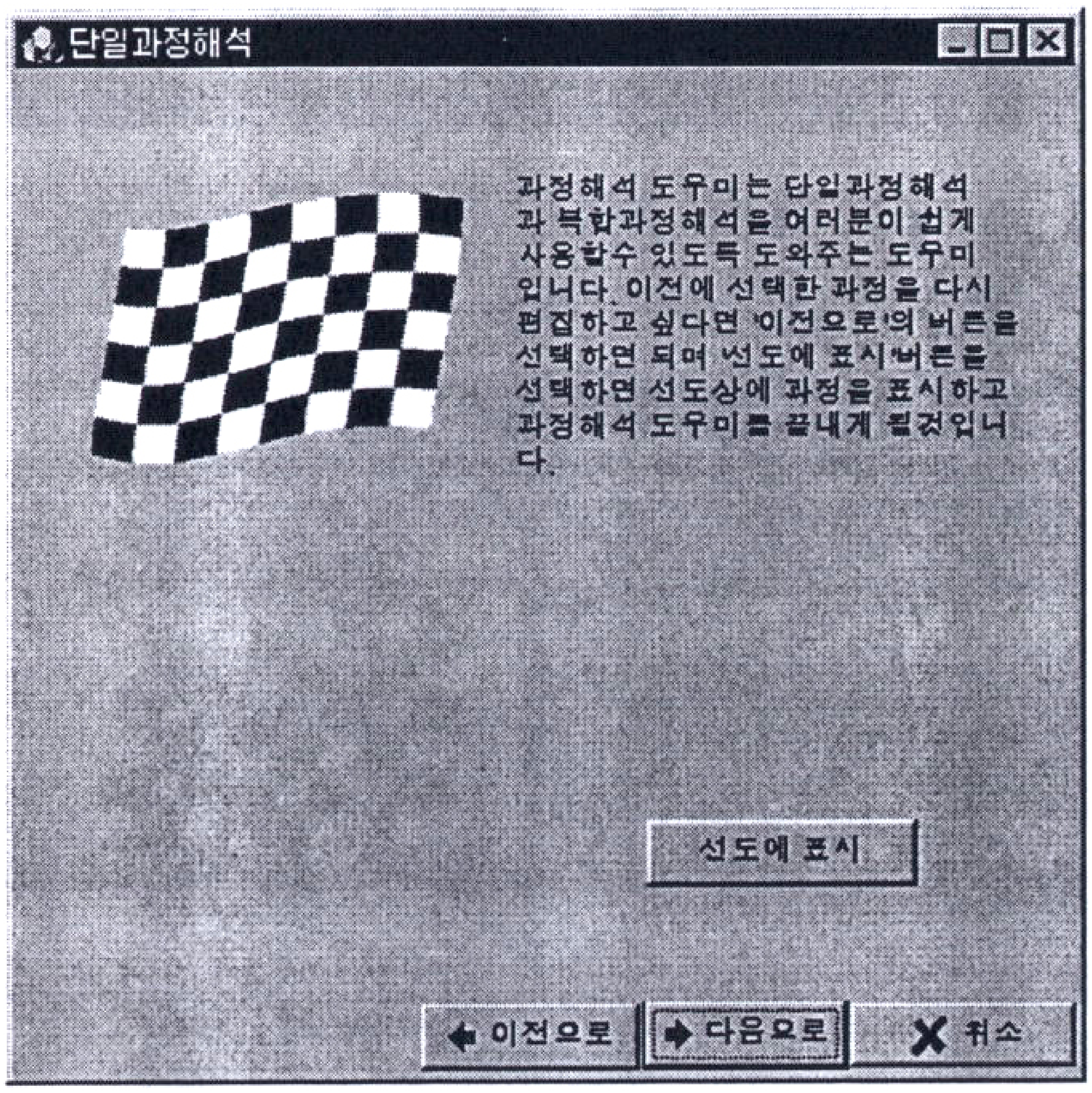


2. 계산결과 출력 화면



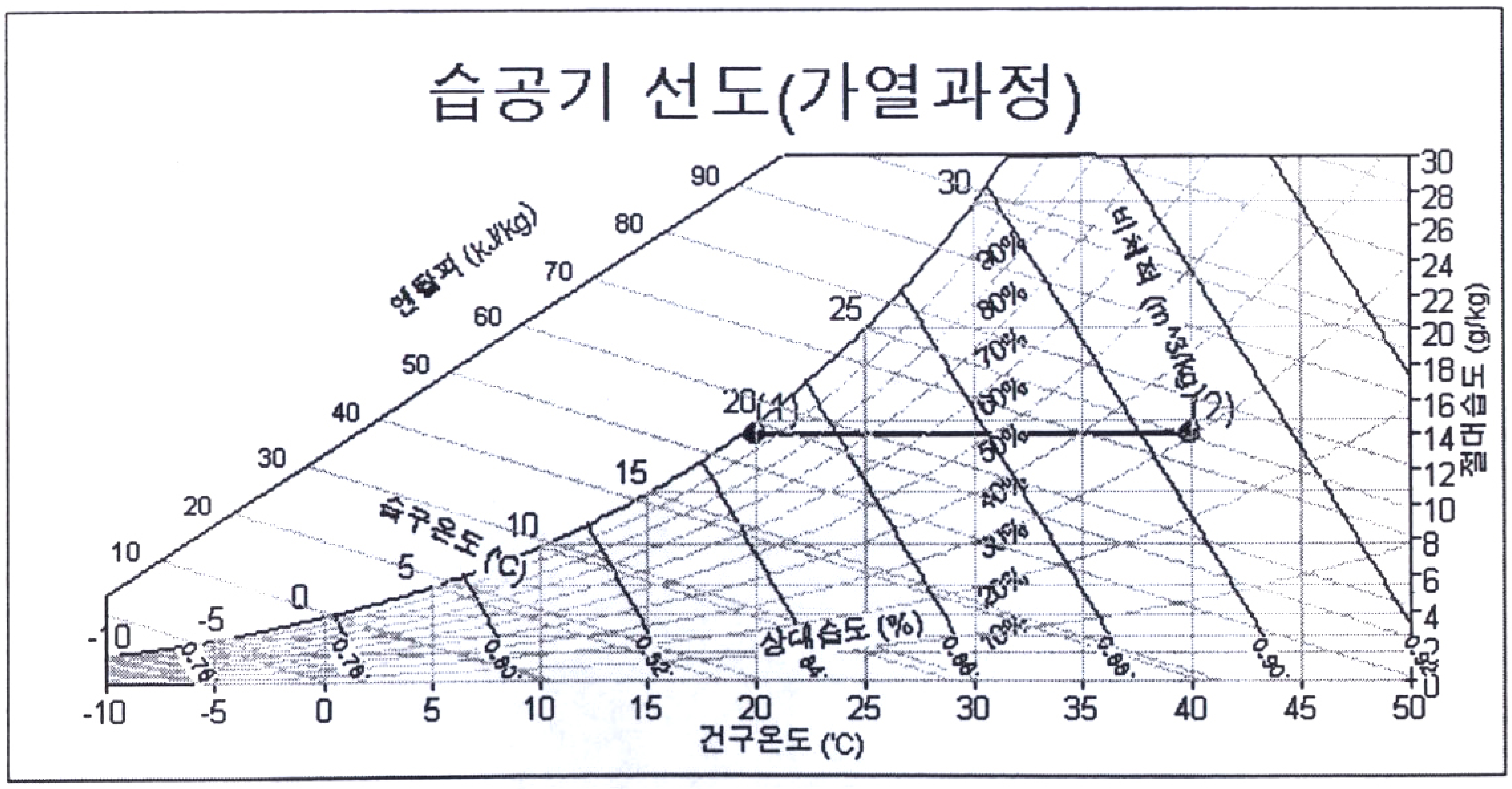
3. 습공기선도 작성여부 선택 화면

다음의 화면에서 "선도에 표시" 버튼을 누르면 습공기선도가 작성된다.



4. 습공기선도 작성 화면

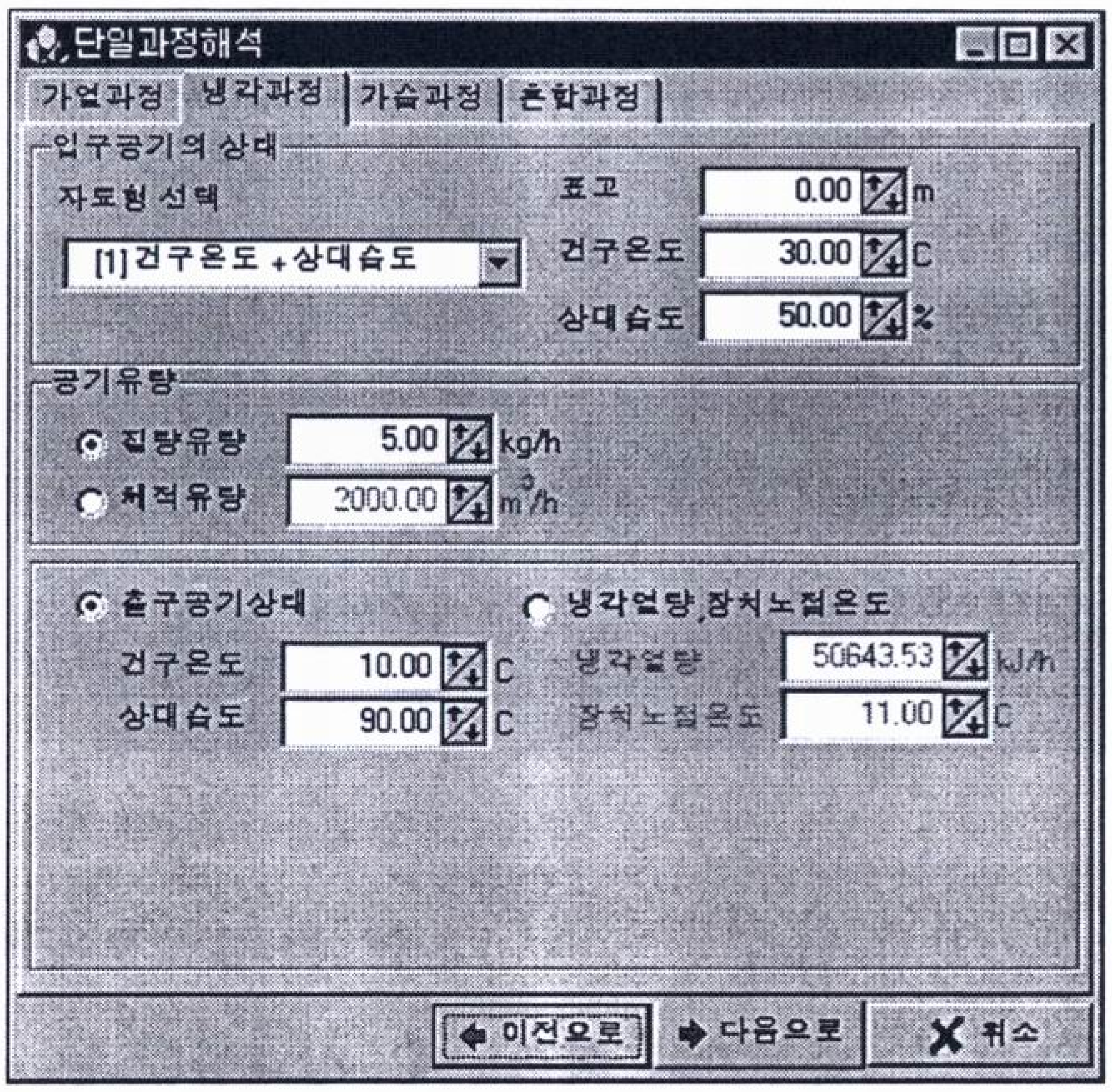
다음의 선도에서 상태점(1)은 습공기의 최초상태, 상태점(2)는 습공기의 최종상태를 나타낸다.



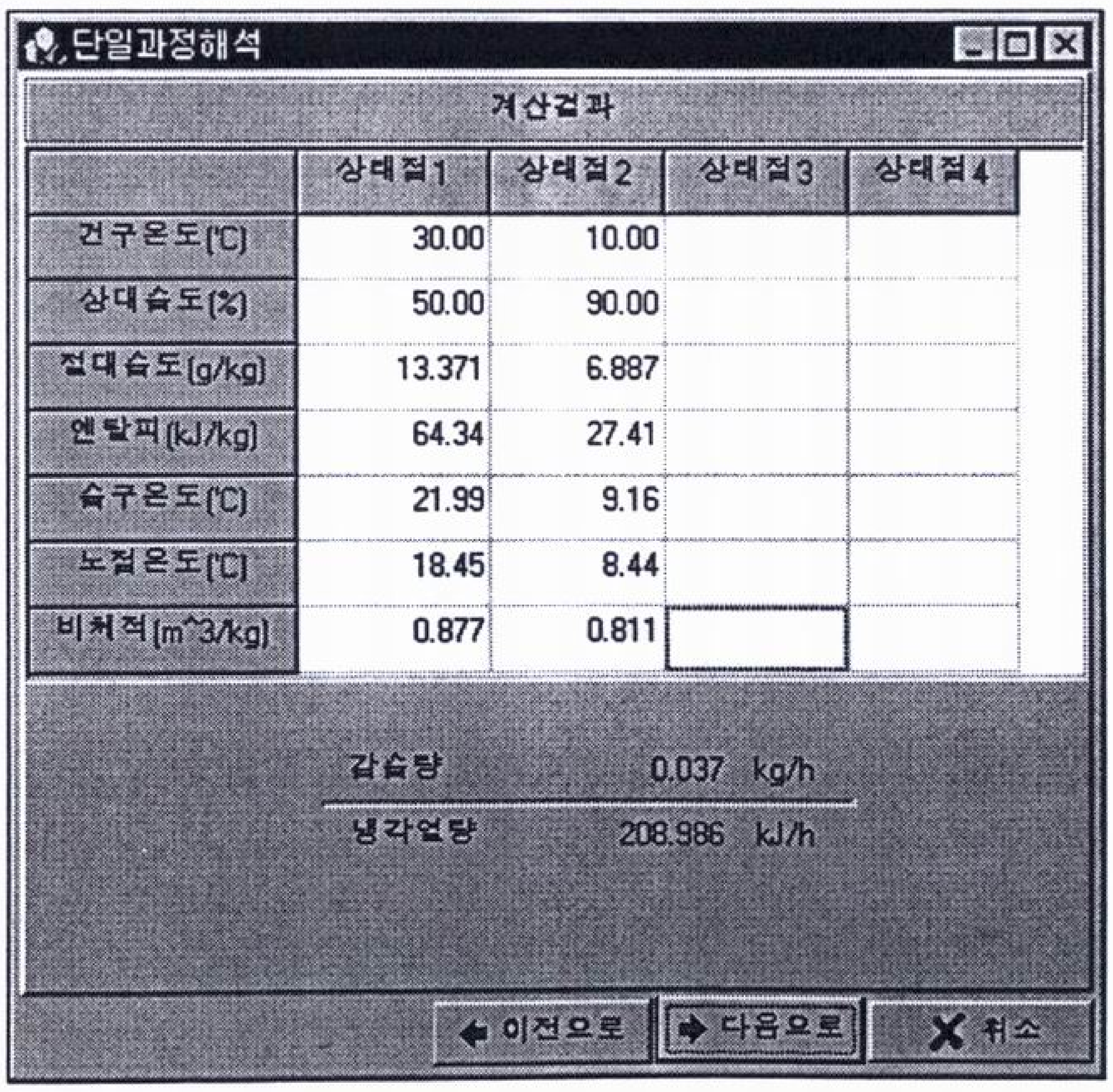
(나) 냉각과정 해석

사용 예로서 표고가 0.0 ㎙인 지역에서 입구공기의 상태가 건구온도 30℃, 상태습도 50%인 습공기 5 ㎥/s를 건구온도 10℃, 상대습도 90%까지 냉각감습할 경우에 소요되는 냉각열량과 이때 발생하는 감습량을 계산하고 과정해석을 습공기 선도로 나타내면 다음과 같다.

1. 입력화면



2. 계산결과 출력화면


단일과정해석

계산결과

	상태점1	상태점2	상태점3	상태점4
건구온도[℃]	30.00	10.00		
상대습도[%]	50.00	90.00		
절대습도[g/kg]	13.371	6.887		
엔탈피[kJ/kg]	64.34	27.41		
습구온도[℃]	21.99	9.16		
노점온도[℃]	18.45	8.44		
비체적[m^3/kg]	0.877	0.811		

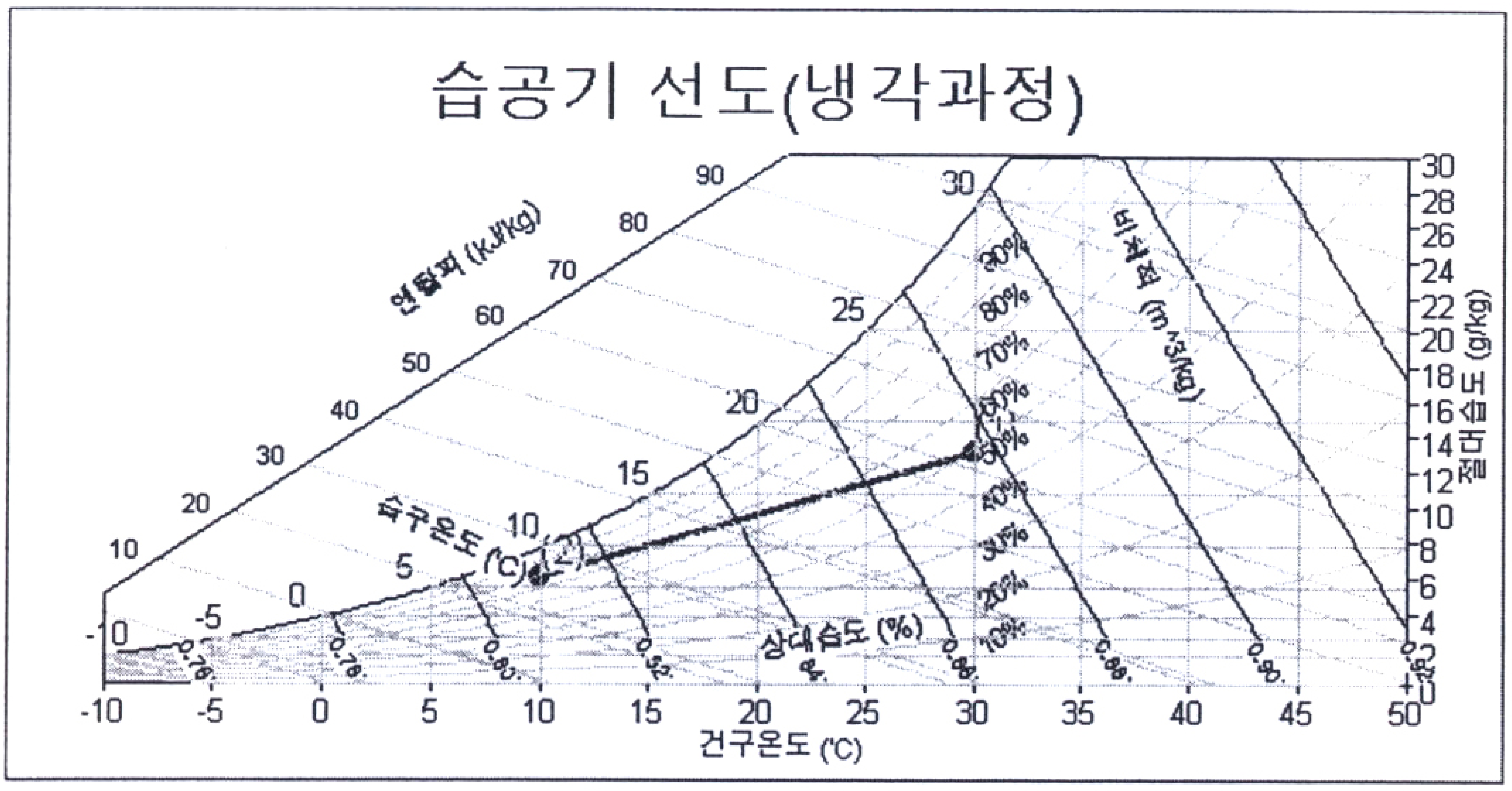
감습량 0.037 kg/h

냉각열량 208.986 kJ/h

이전으로 다음으로 취소

3. 습공기선도 작성 화면

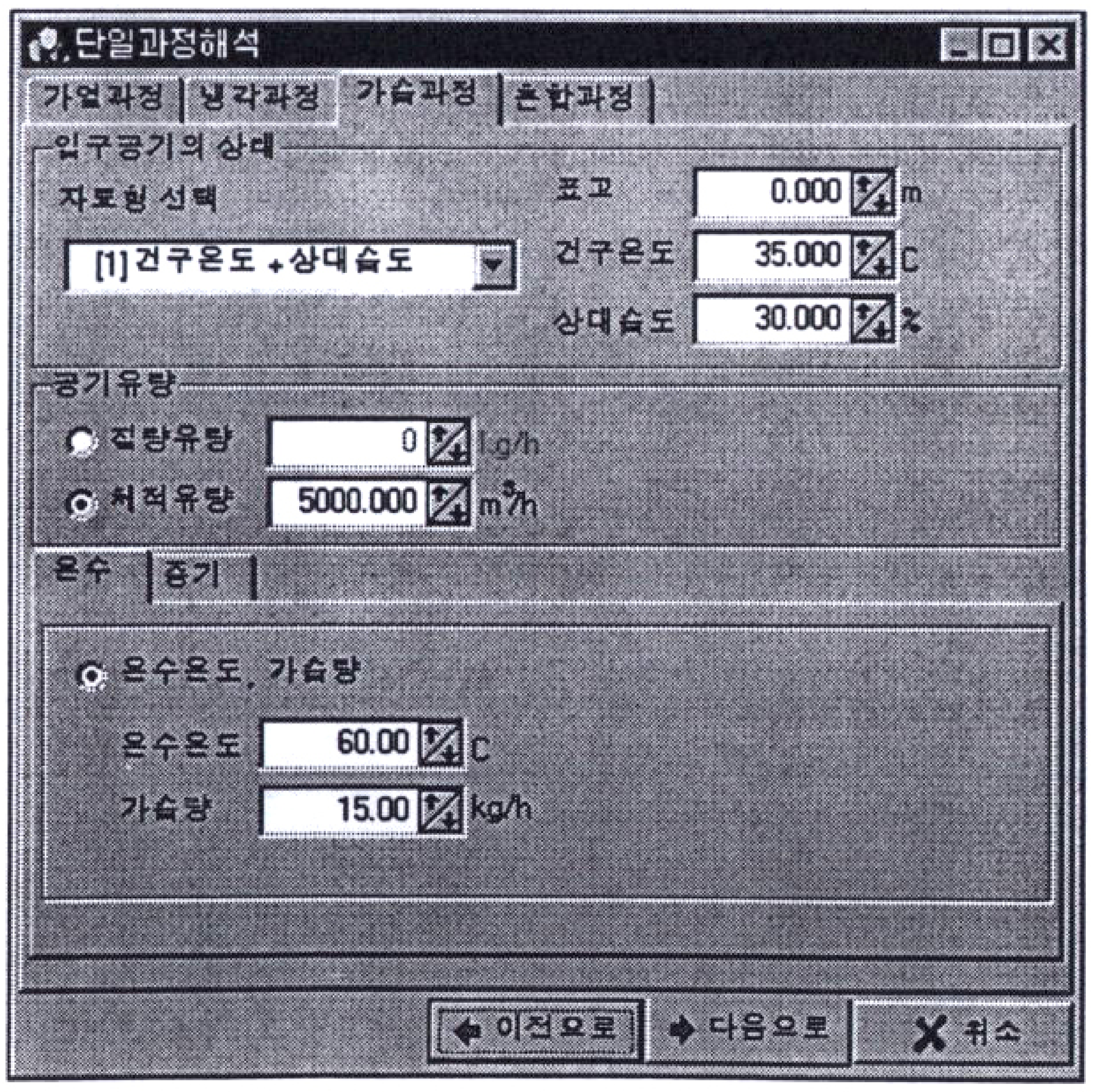
다음의 선도에서 상태점(1)은 습공기의 최초상태, 상태점(2)는 습공기의 최종상태를 나타낸다.



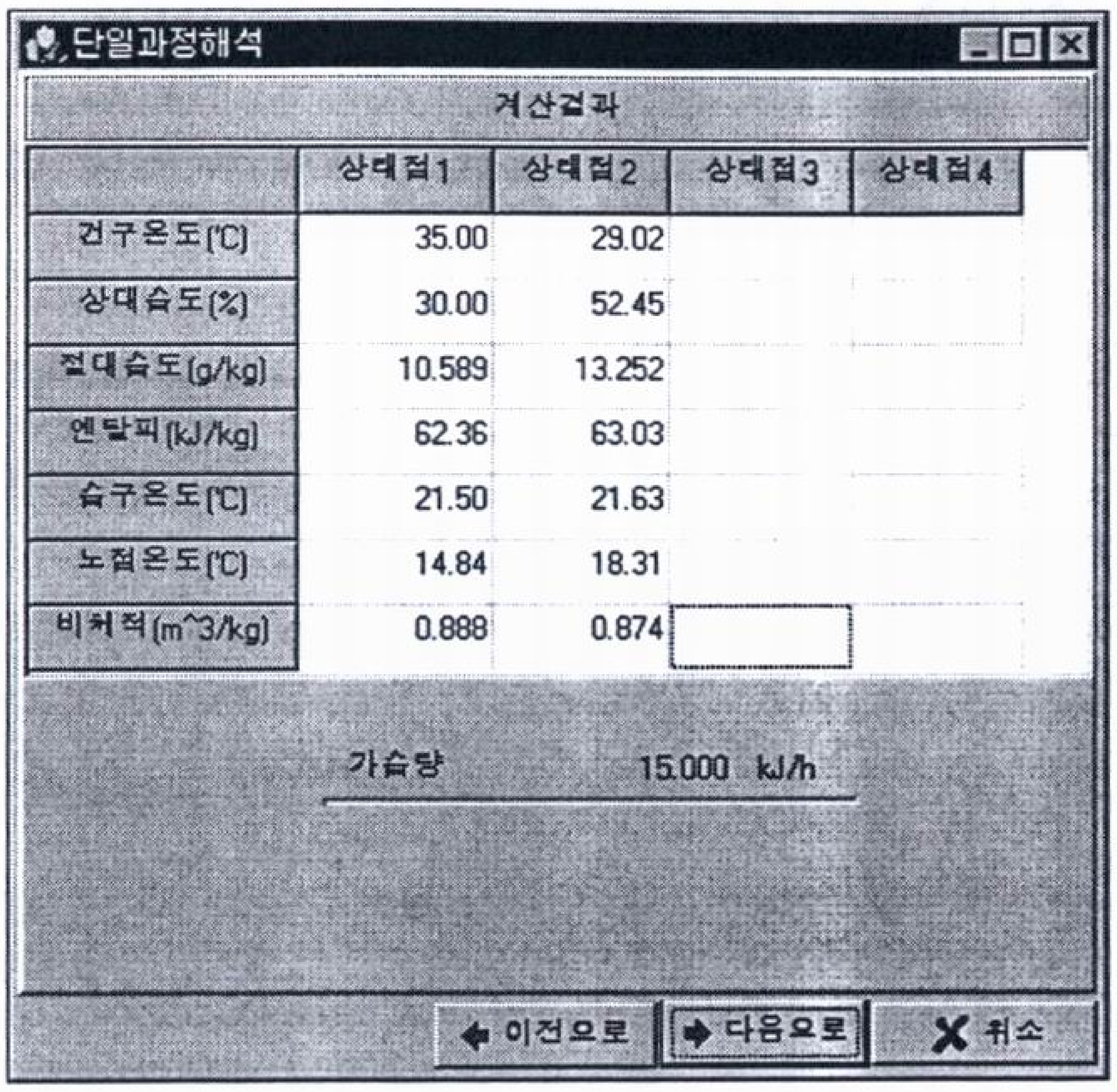
(다) 가습과정 해석

가습과정은 가습방법에 따라 온수 및 증기로 나누어 계산할 수 있도록 되어 있다. 사용 예로서 온수가습의 경우를 보면 표고가 0.0 m인 지역에서 건구온도 35℃, 상태습도 30%인 공기 5000 ㎥/h를 60℃의 온수로 15 kg/h로 가습할 경우에 가습된 공기의 상태를 계산하고 과정해석을 습공기선도에 나타내면 다음과 같다.

1. 입력화면



2. 계산결과 출력화면



단일과정해석

계산결과

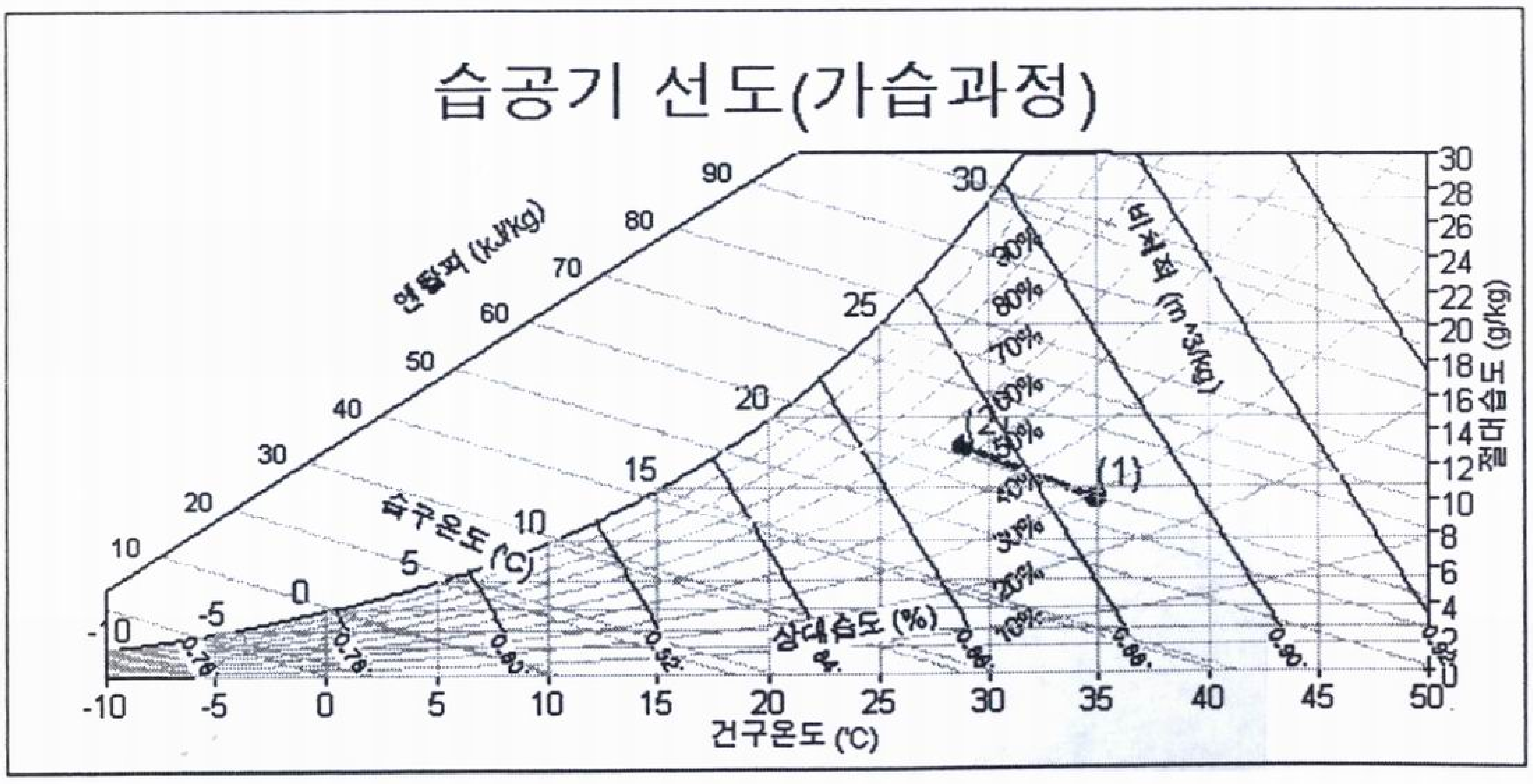
	상태점1	상태점2	상태점3	상태점4
건구온도(℃)	35.00	29.02		
상대습도(%)	30.00	52.45		
절대습도(g/kg)	10.589	13.252		
엔탈피(kJ/kg)	62.36	63.03		
습구온도(℃)	21.50	21.63		
노점온도(℃)	14.84	18.31		
비체적(m^3/kg)	0.888	0.874		

가습량 15.000 kJ/h

이전으로 다음으로 취소

3. 습공기선도 작성 화면

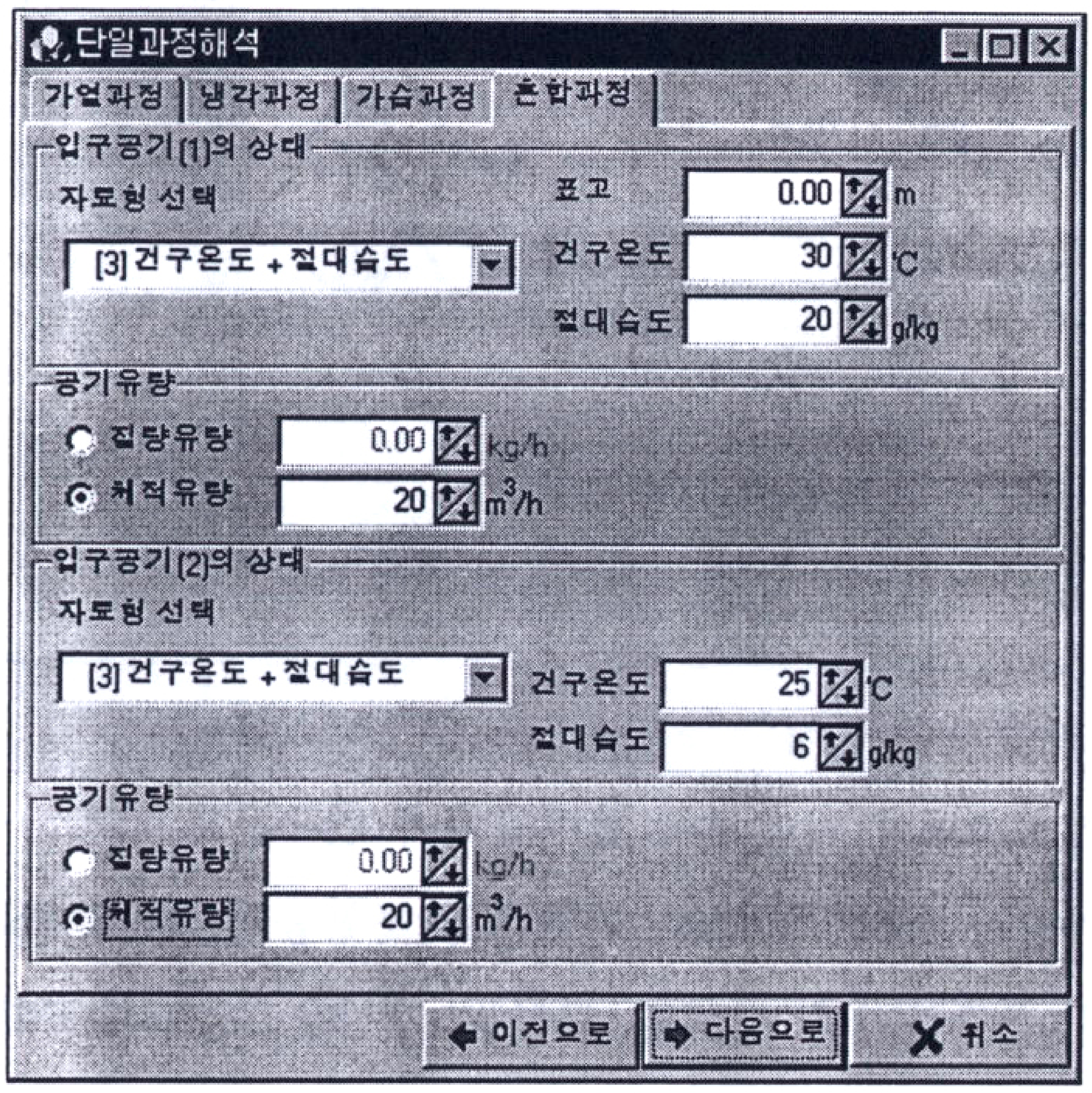
다음의 선도에서 상태점(1)은 습공기의 최초상태, 상태점(2)는 습공기의 최종상태를 나타낸다.



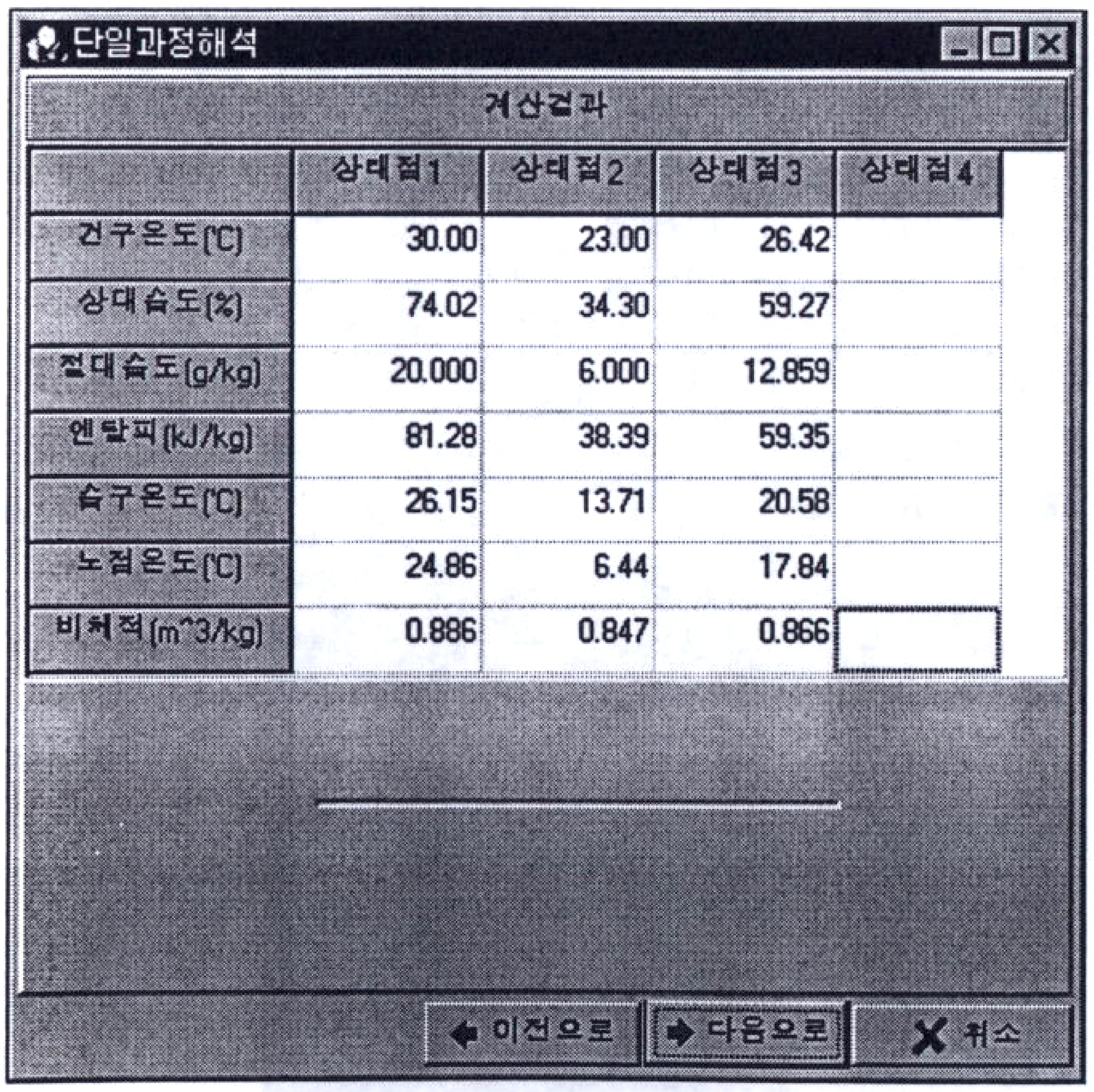
(라) 혼합과정 해석

사용 예로서 표고가 0.0 m인 지역에서 건구온도 30℃, 절대습도 20 g/kg인 공기 20 ㎥/h와 건구온도 25℃, 절대습도 6 g/kg인 공기 20 ㎥/h을 혼합할 때의 혼합점의 상태량을 계산하고 습공기선도에 나타내면 다음과 같다.

1. 입력화면



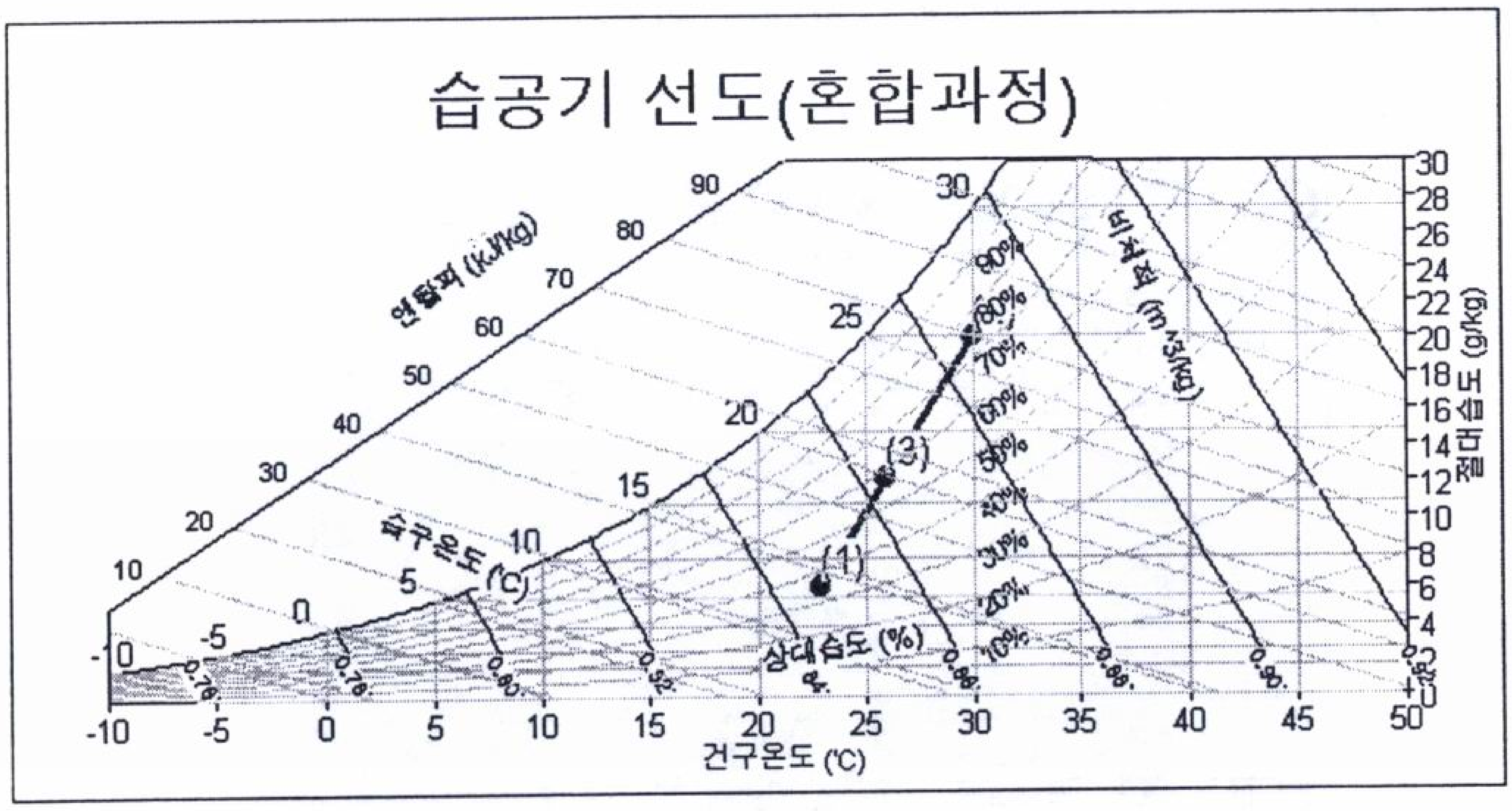
2. 계산결과 출력화면



	상태점1	상태점2	상태점3	상태점4
건구온도[℃]	30.00	23.00	26.42	
상대습도[%]	74.02	34.30	59.27	
절대습도[g/kg]	20.000	6.000	12.859	
엔탈피[kJ/kg]	81.28	38.39	59.35	
습구온도[℃]	26.15	13.71	20.58	
노점온도[℃]	24.86	6.44	17.84	
비체적[m^3/kg]	0.886	0.847	0.866	

3. 습공기선도 작성 화면

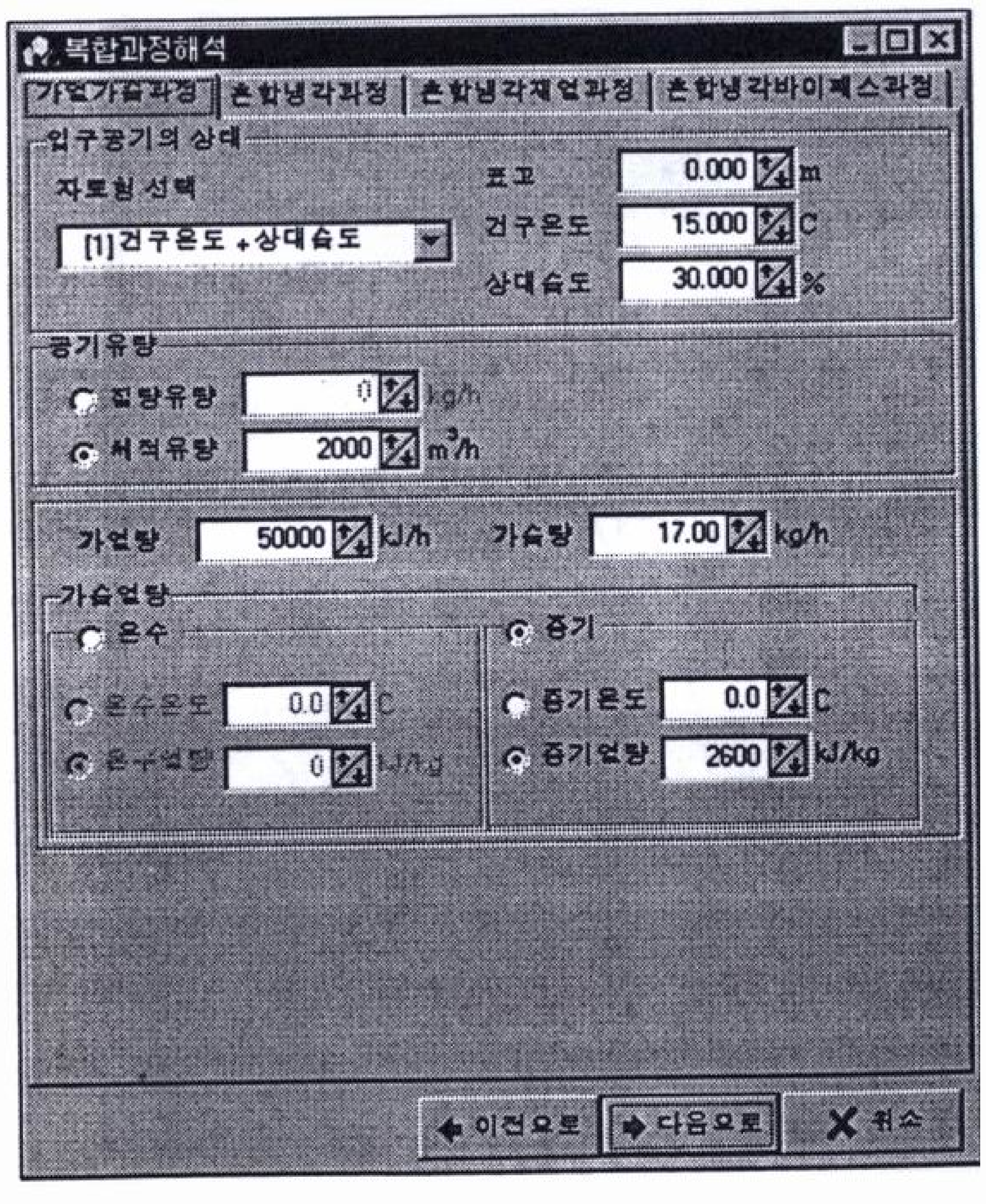
다음 선도에서 상태점(1)과 (2)는 습공기의 최초상태, 상태점(3)은 습공기의 혼합상태를 나타낸다.



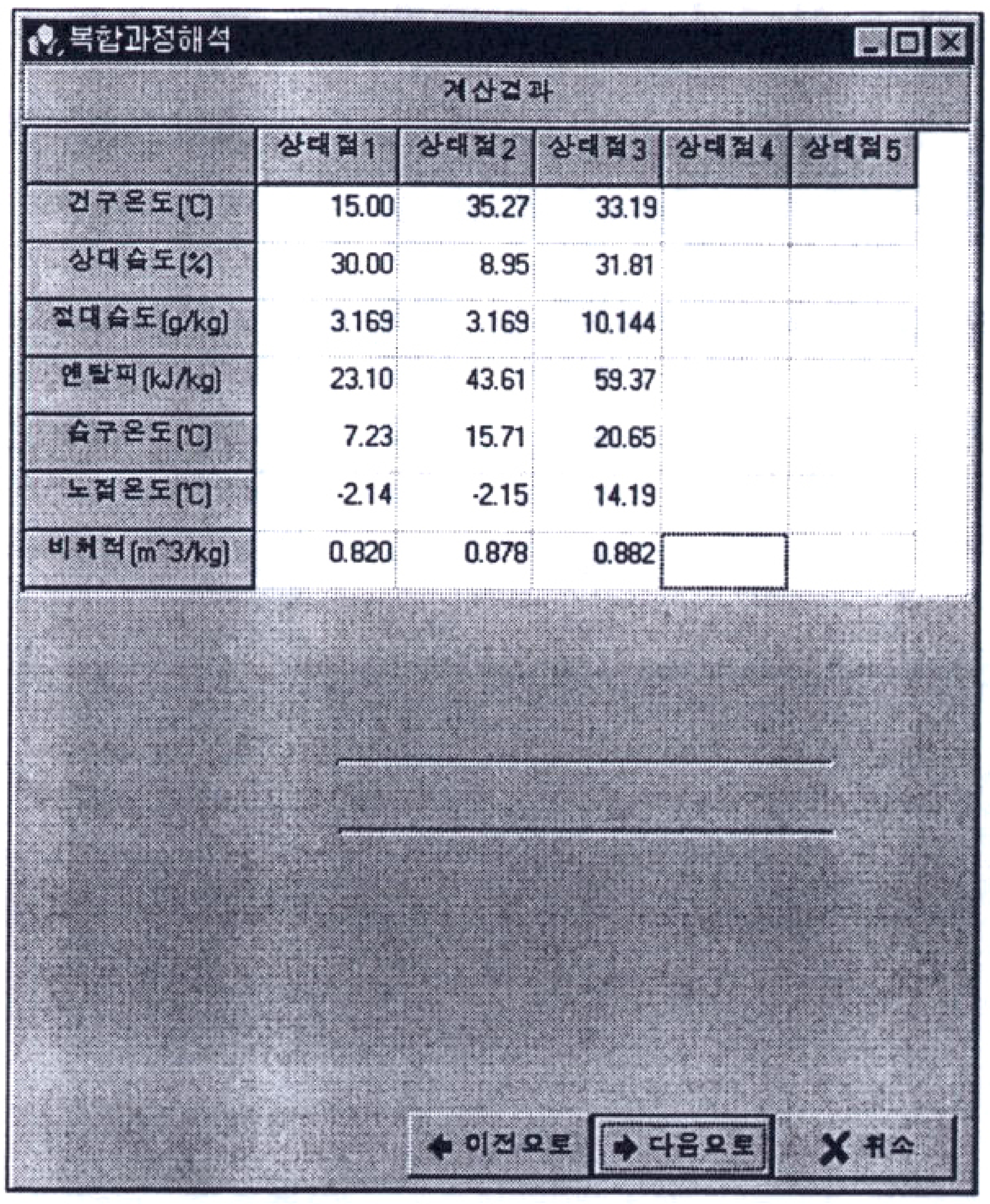
(마) 가열가습과정 해석

사용예로서 표고가 0.0 m의 지역에서 입구공기가 건구온도 15℃, 상대습도 30%인 공기 2000 ㎥/h에 가열량 45000 kJ/h로 가열하고 포화증기(열량, 2600 kJ/kg)로 17 kg/h을 가습한 후의 출구공기의 상태를 구하는 계산을 하고 습공기의 과정해석을 하면 다음과 같다.

1. 입력 화면



2. 계산결과 출력화면

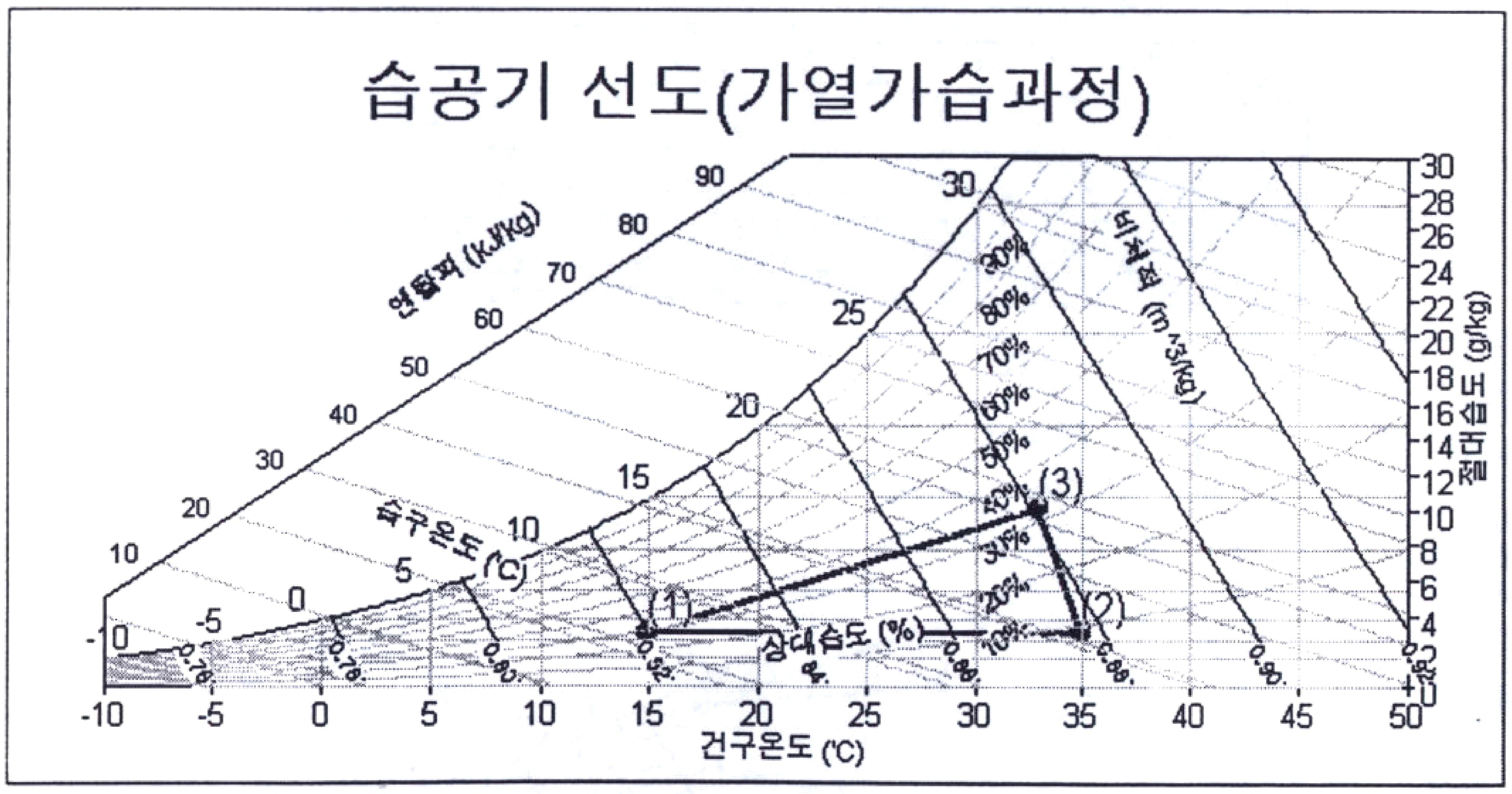

복합과정해석

계산결과

	상태점1	상태점2	상태점3	상태점4	상태점5
건구온도(℃)	15.00	35.27	33.19		
상대습도(%)	30.00	8.95	31.81		
절대습도(g/kg)	3.169	3.169	10.144		
엔탈피(kJ/kg)	23.10	43.61	59.37		
습구온도(℃)	7.23	15.71	20.65		
노점온도(℃)	-2.14	-2.15	14.19		
비체적(m^3/kg)	0.820	0.878	0.882		

3. 습공기선도 작성 화면

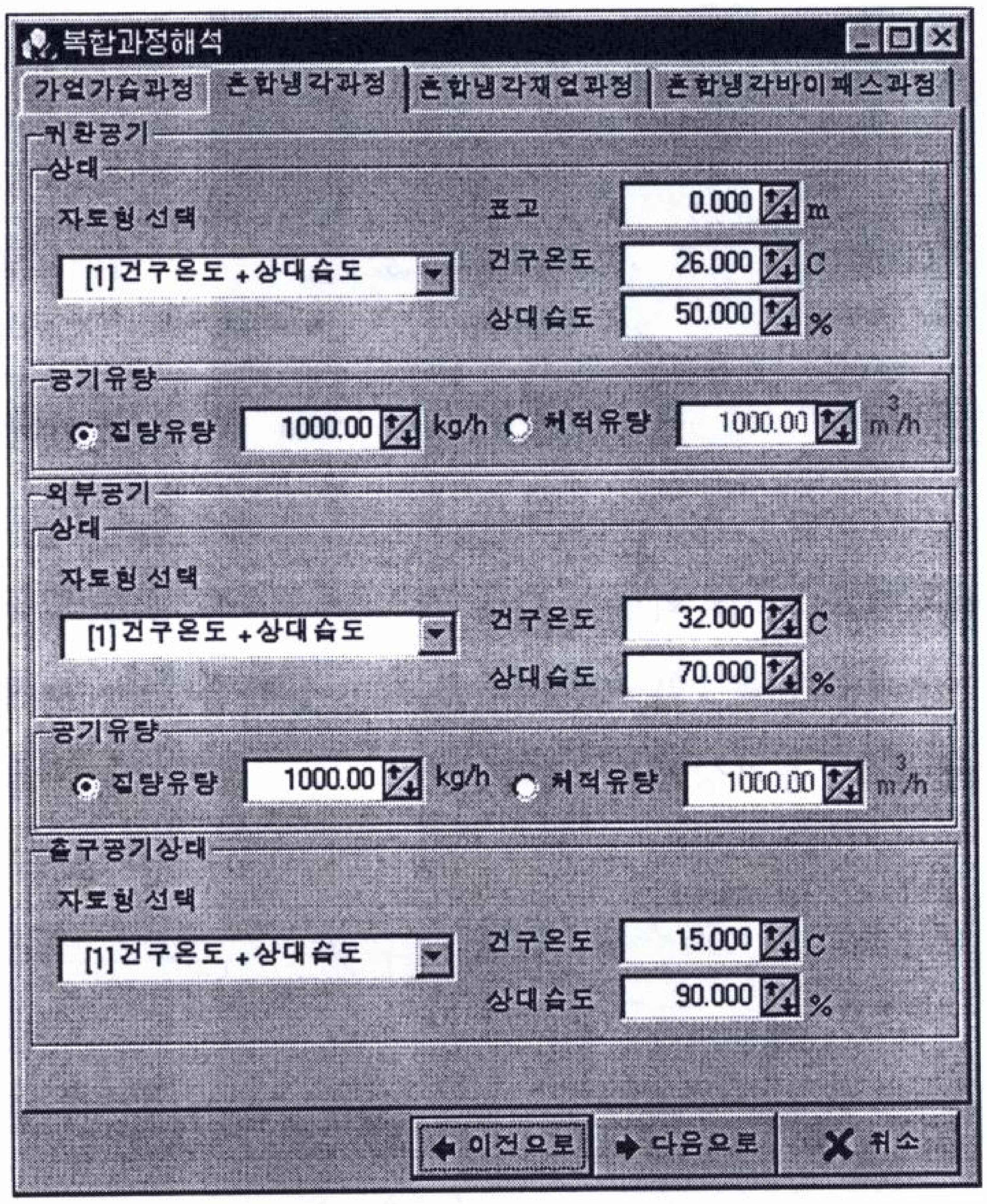
다음의 선도에서 상태점(1)은 입구공기의 상태, 상태점(2)는 가열된 공기의 상태, 상태점(3)은 가습후의 공기상태를 나타낸다.



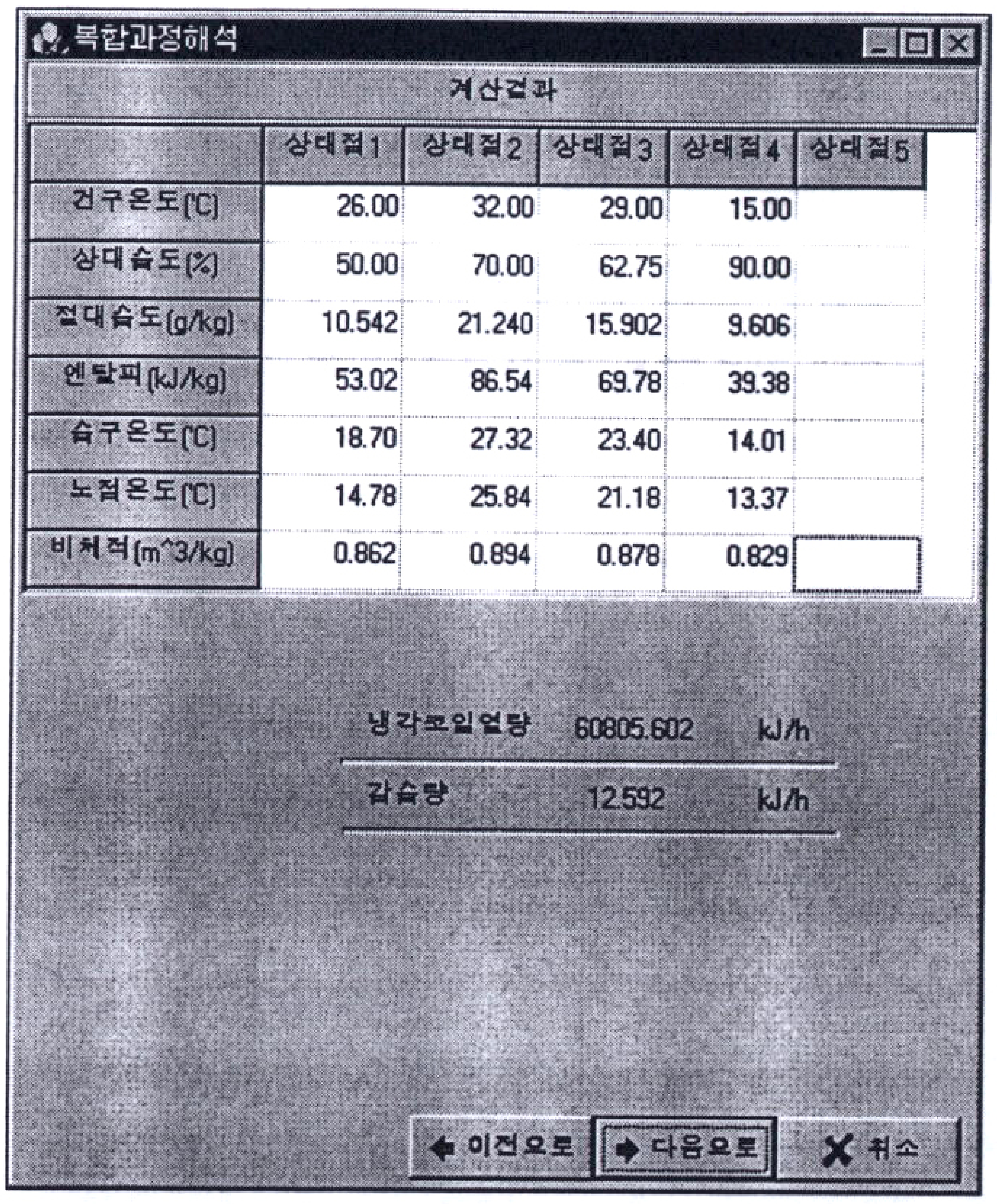
(바) 혼합냉각과정 해석

사용 예로서 표고가 0.0 m인 지역에서 실내 귀환공기가 건구온도 26℃, 상대습도 50%인 공기 1000 kg/h와 외부공기의 건구온도 32℃, 상대습도 60%인 공기 1000 kg/h을 혼합한후에 냉각감습하여 출구공기의 상태가 건구온도 15℃, 상대습도 70%가 되도록 할 때의 혼합점의 상태와 냉각열량 및 감습량을 계산하고 습공기의 과정 해석을 하면 다음과 같다.

1. 입력화면



2. 계산결과 출력화면



복합과정해석

계산결과

	상태점1	상태점2	상태점3	상태점4	상태점5
건구온도(℃)	26.00	32.00	29.00	15.00	
상대습도(%)	50.00	70.00	62.75	90.00	
절대습도(g/kg)	10.542	21.240	15.902	9.606	
엔탈피(kJ/kg)	53.02	86.54	69.78	39.38	
습구온도(℃)	18.70	27.32	23.40	14.01	
노점온도(℃)	14.78	25.84	21.18	13.37	
비체적(m^3/kg)	0.862	0.894	0.878	0.829	

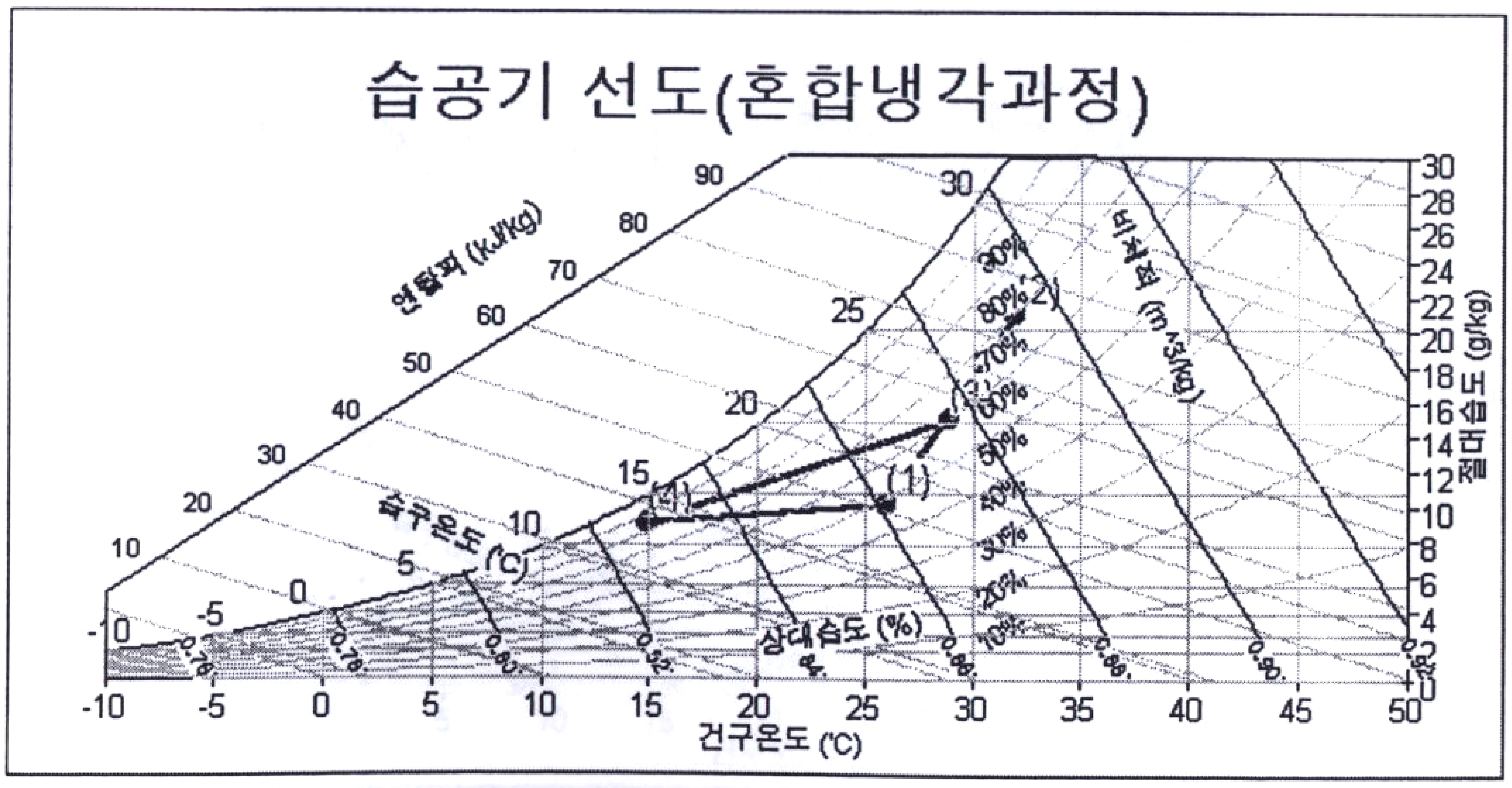
냉각코일열량 60805.602 kJ/h

감습량 12.592 kJ/h

이전으로 다음으로 취소

3. 습공기선도 작성 화면

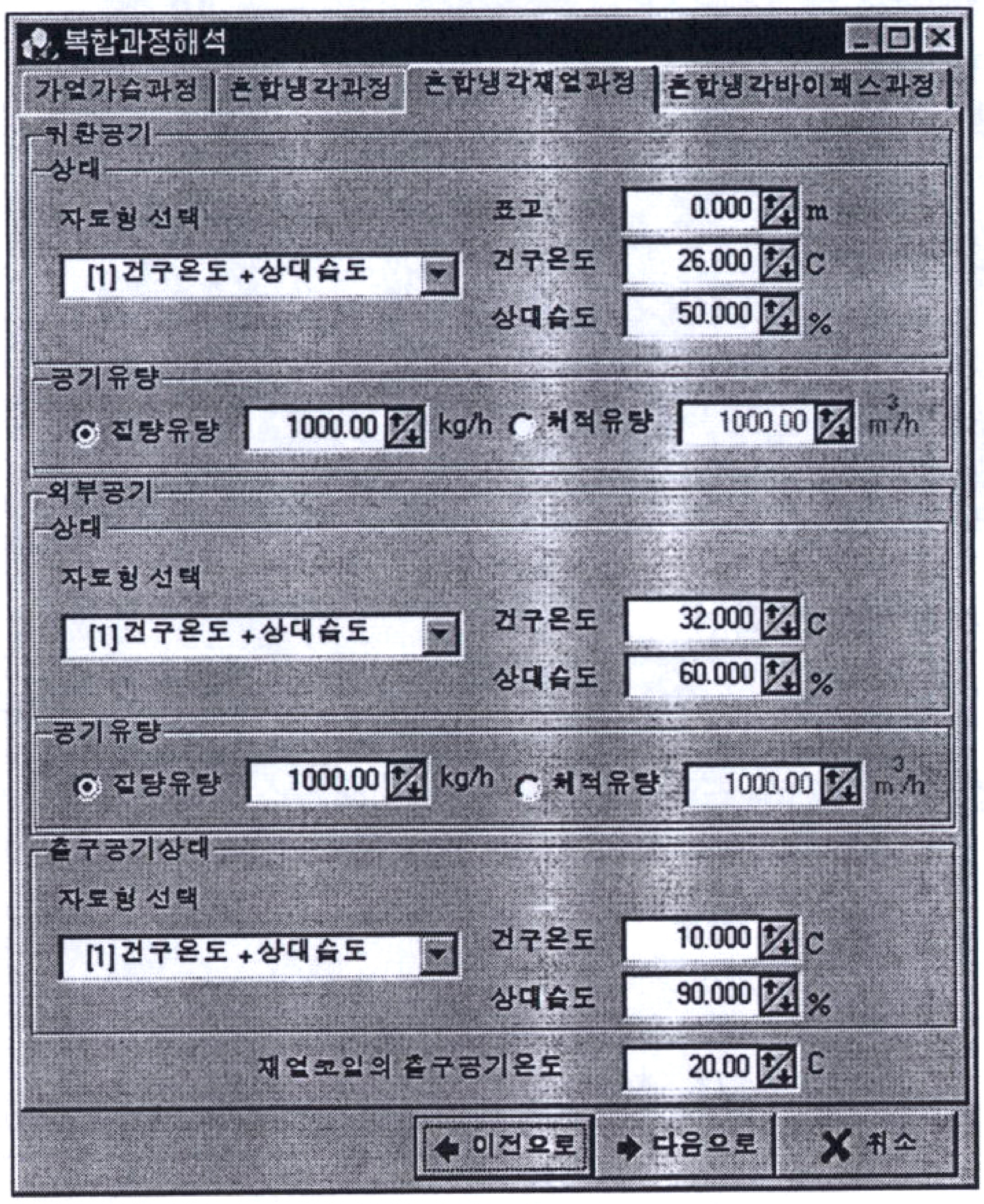
다음의 선도에서 상태점(1)은 실내공기의 상태, 상태점(2)는 외부공기의 상태, 상태점(3)은 상태점(1), (2)의 공기를 혼합한 상태, 상태점(4)는 혼합후 냉각감습된 공기의 상태를 나타낸다.



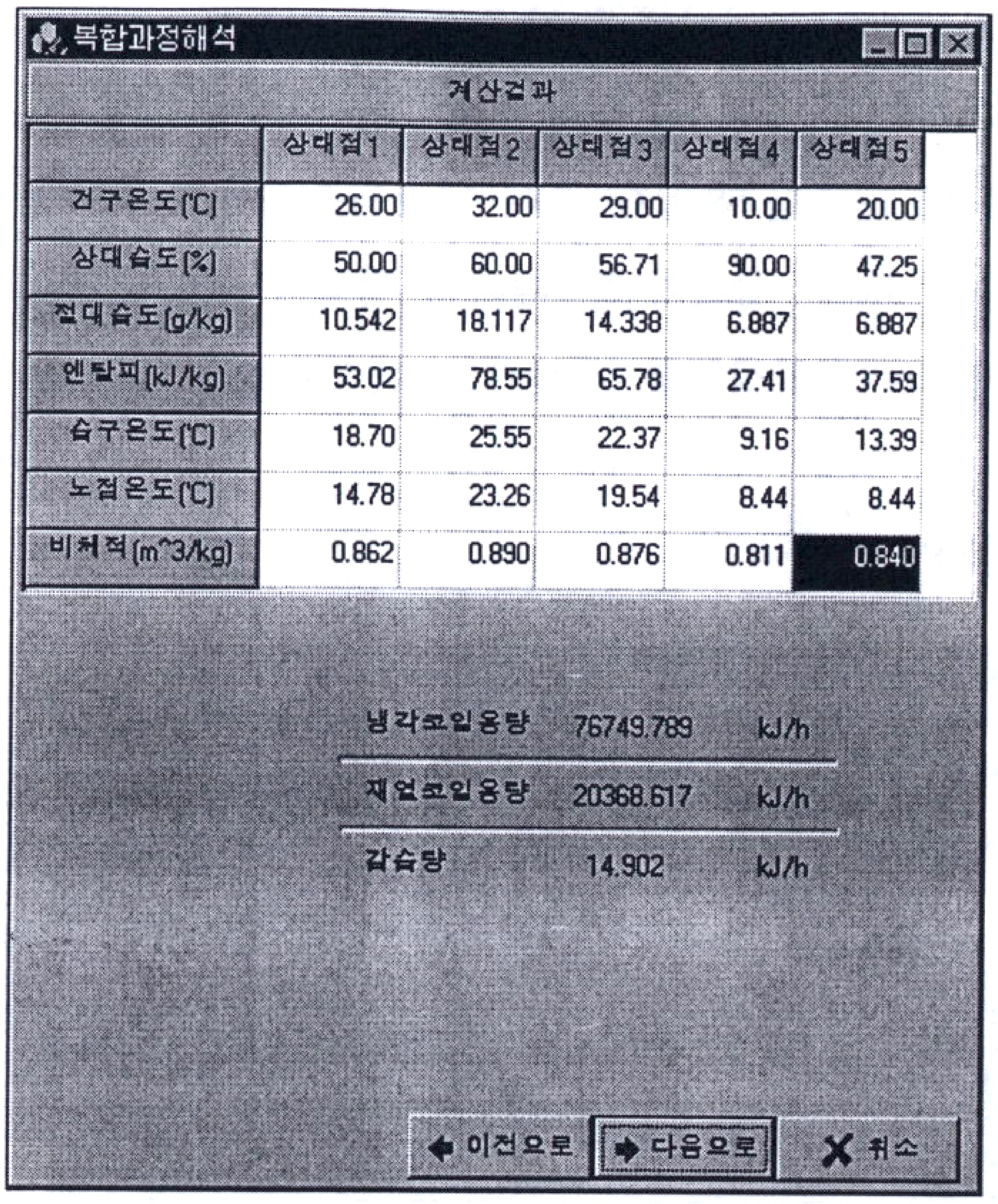
(사) 혼합냉각 및 재열과정 해석

사용 예로서 표고가 0.0 ㎙인 지역에서 실내 귀환공기의 건구온도 26℃, 상대습도 50%인 공기 1000 kg/h과 외부공기의 건구온도 32℃, 상대습도 60%인 공기 1000 kg/h을 혼합한 후에 냉각감습하여 건구온도 10℃, 상대습도 90%인 공기를 재열하여 20℃의 공기로 만들때에 혼합점과 냉각코일출구 및 재열코일출구의 공기 상태점과 냉각코일열량, 재열코일열량, 감습량을 계산하고 과정해석을 습공기선도에 나타내면 다음과 같다.

1. 입력화면



2. 계산결과 출력화면


복합과정해석

계산결과

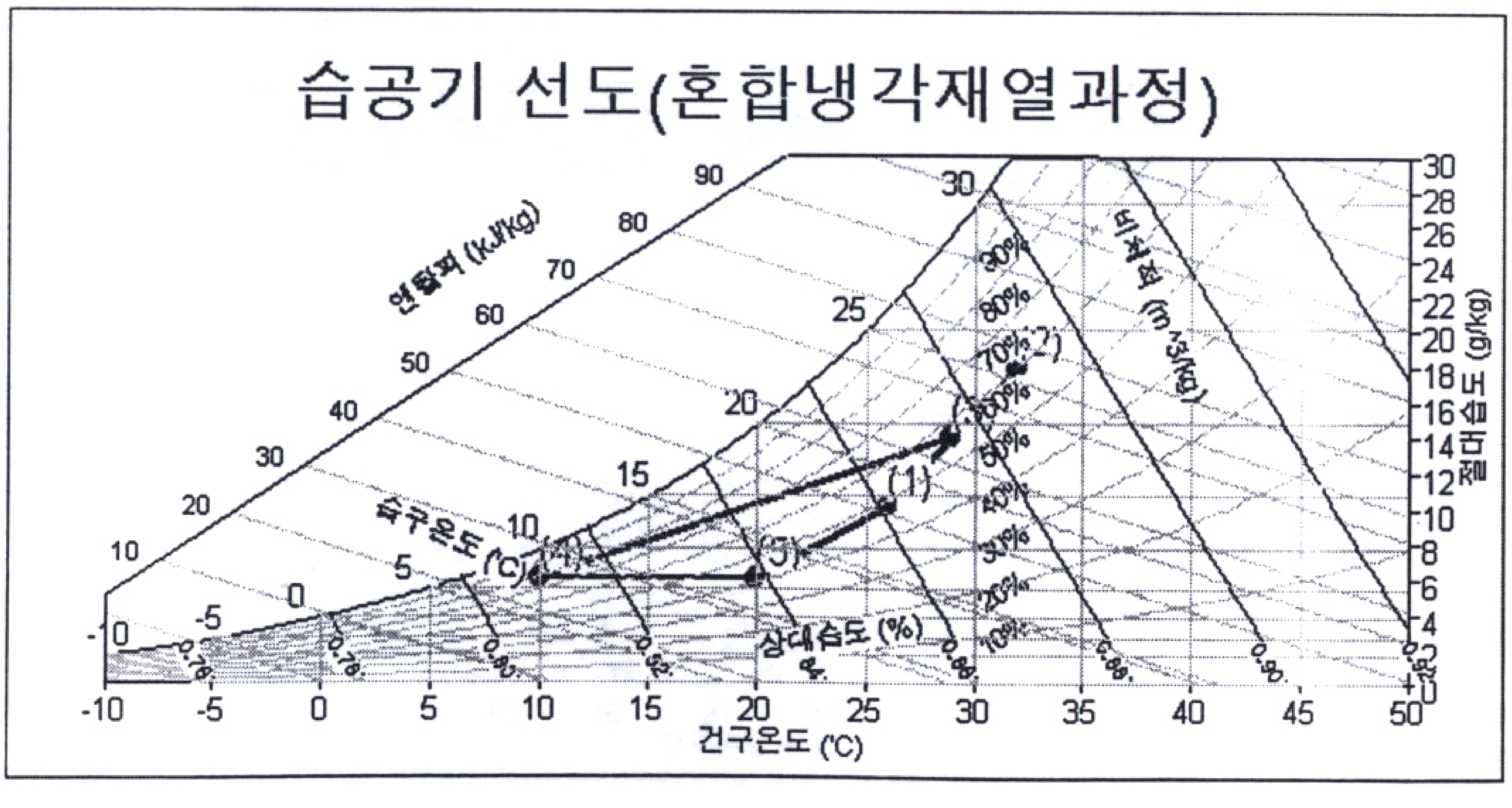
	상태점1	상태점2	상태점3	상태점4	상태점5
건구온도(℃)	26.00	32.00	29.00	10.00	20.00
상대습도(%)	50.00	60.00	56.71	90.00	47.25
절대습도(g/kg)	10.542	18.117	14.338	6.887	6.887
엔탈피(kJ/kg)	53.02	78.55	65.78	27.41	37.59
습구온도(℃)	18.70	25.55	22.37	9.16	13.39
노점온도(℃)	14.78	23.26	19.54	8.44	8.44
비체적(m^3/kg)	0.862	0.890	0.876	0.811	0.840

냉각코일용량	76749.789	kJ/h
재열코일용량	20368.617	kJ/h
감습량	14.902	kJ/h

이전으로 다음으로 취소

3. 습공기선도 작성 화면

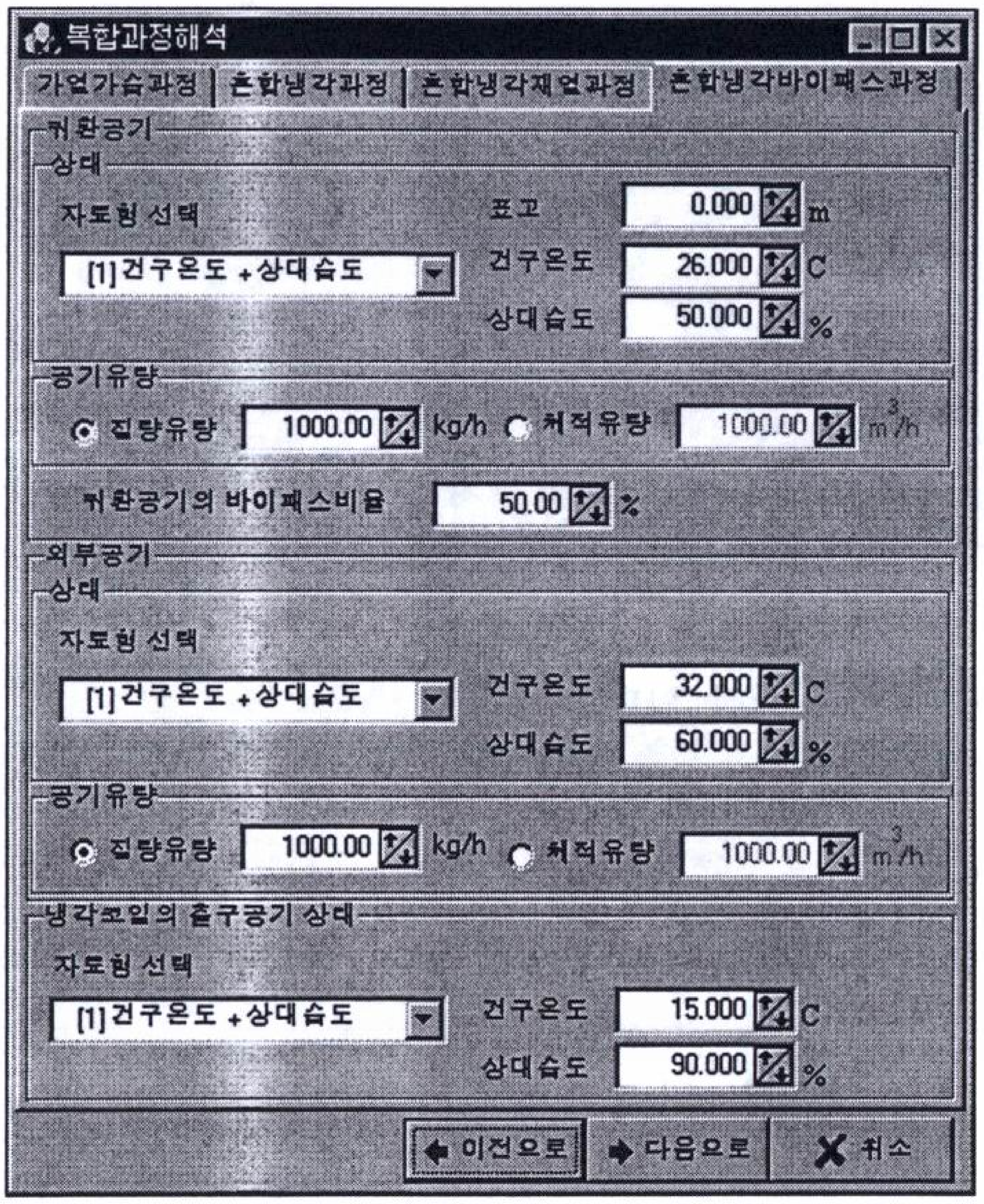
다음의 선도에서 상태점(1)은 실내공기 상태, 상태점(2)는 외부공기상태, 상태점(3)은 실내와 외부의 공기를 혼합한 상태, 상태점(4)는 혼합된 공기를 냉각감습한 후의 상태, 상태점(4)는 상태점(3)을 가열한 후의 상태를 나타낸다.



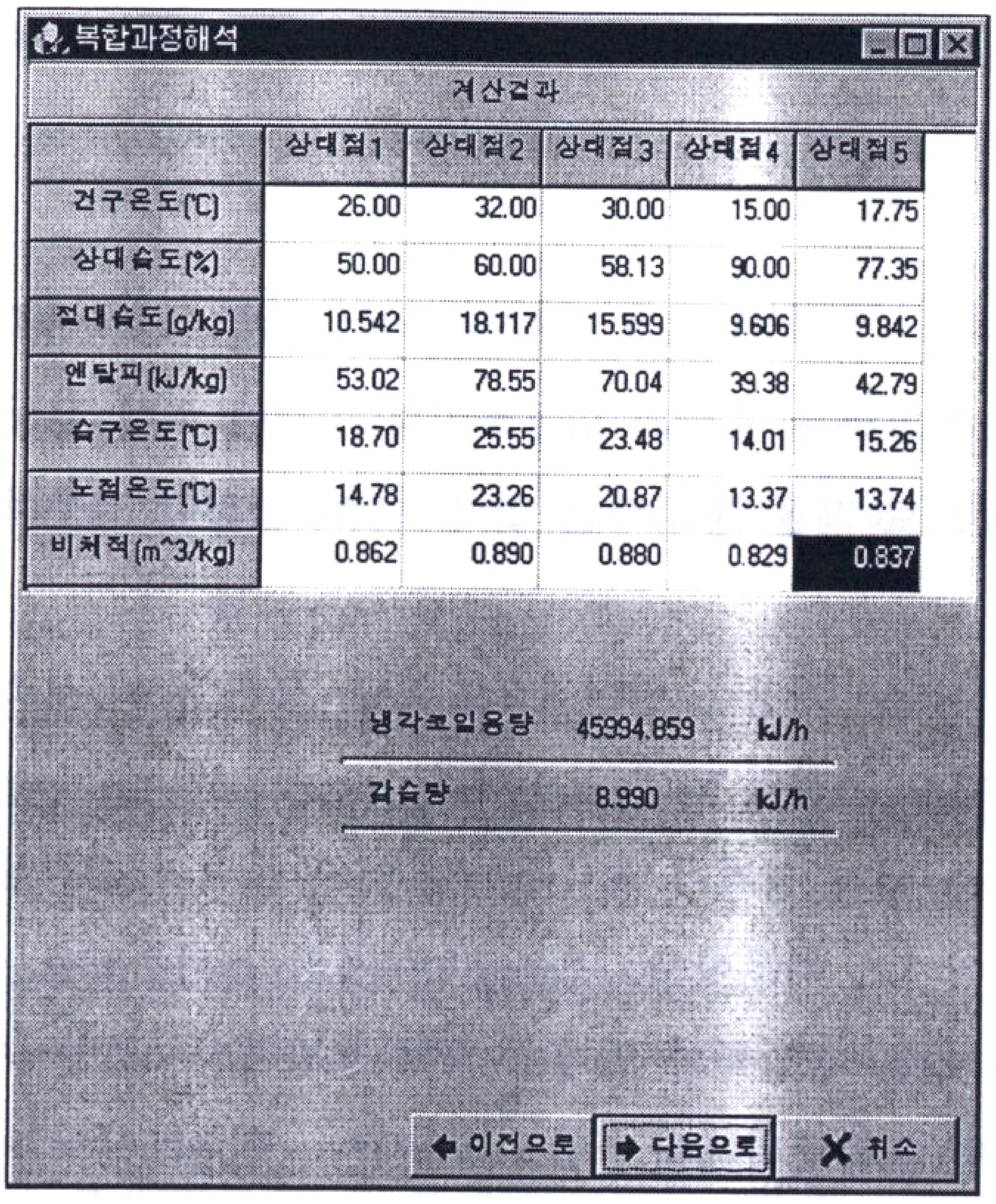
(아) 혼합냉각 및 바이패스과정 해석

사용 예로서 표고가 0.0 m인 지역에서 실내 귀환공기가 건구온도 26℃, 상대습도 50%인 공기 1000 kg/h이고 바이패스비율이 50%이며, 외부공기의 건구온도 32℃, 상대습도 60%인 공기 1000 kg/h을 혼합한 후에 냉각감습하여 냉각코일 출구공기가 건구온도 15℃, 상대습도 90%인 상태가 될 때의 혼합점의 상태량 및 냉각열량, 감습량 및 바이패스 공기와의 혼합상태를 계산하고 과정해석을 습공기선도에 나타내면 다음과 같다.

1. 입력화면



2. 계산결과의 출력화면



복합과정해석

계산결과

	상대점1	상대점2	상대점3	상대점4	상대점5
건구온도(℃)	26.00	32.00	30.00	15.00	17.75
상대습도(%)	50.00	60.00	58.13	90.00	77.35
절대습도(g/kg)	10.542	18.117	15.599	9.606	9.842
엔탈피(kJ/kg)	53.02	78.55	70.04	39.38	42.79
습구온도(℃)	18.70	25.55	23.48	14.01	15.26
노점온도(℃)	14.78	23.26	20.87	13.37	13.74
비체적(m^3/kg)	0.862	0.890	0.880	0.829	0.837

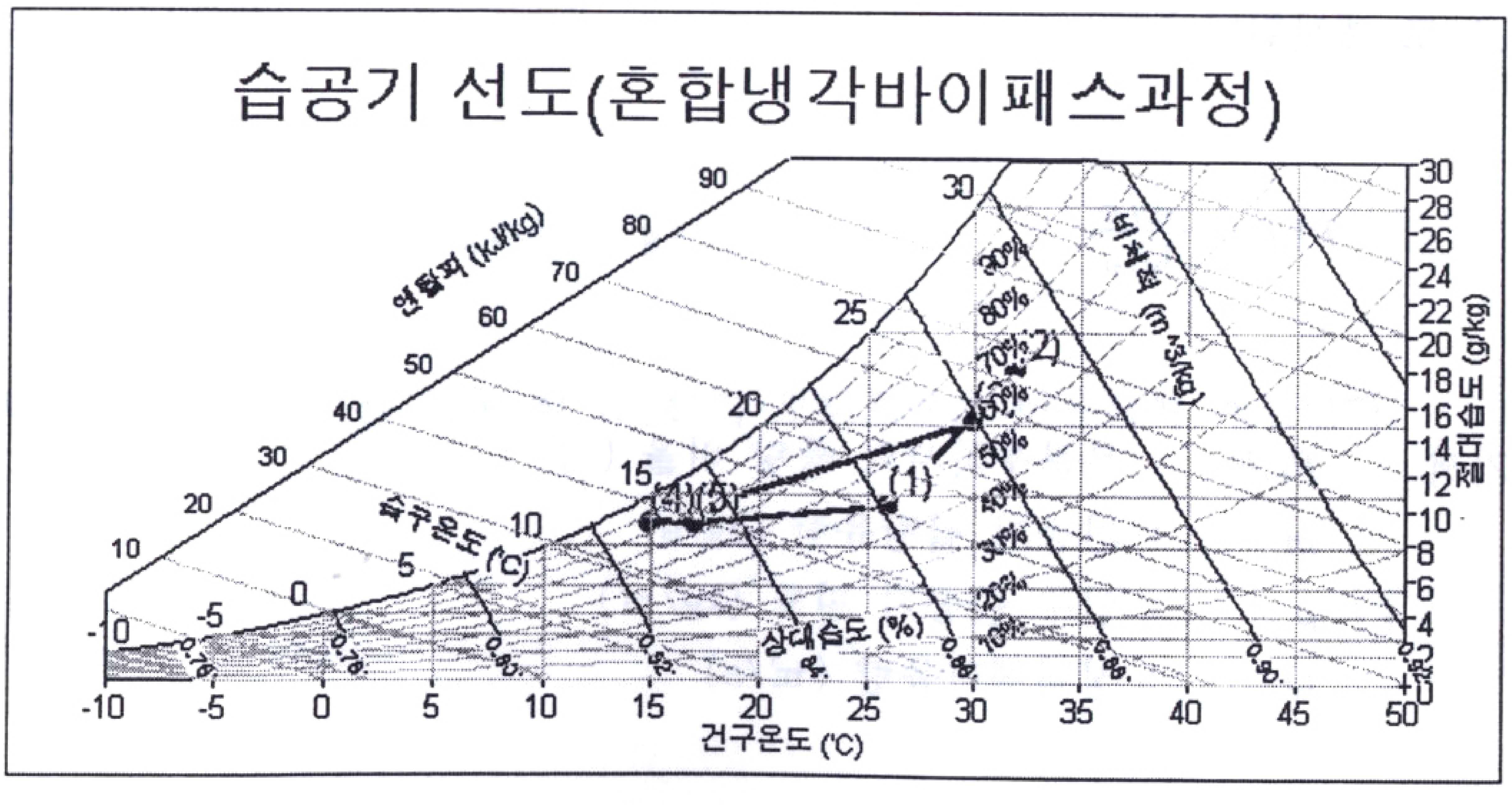
냉각코일용량 45994.859 kJ/h

감습량 8.990 kJ/h

이전으로 다음으로 취소

3. 습공기선도 작성 화면

다음의 선도에서 상태점(1)은 실내공기상태, 상태점(2)는 외부공기상태, 상태점(3)은 실내와 외부의 공기를 혼합한 상태, 상태점(4)는 혼합된 공기를 냉각감습한 후의 상태, 상태점(5)는 상태점(4)의 공기와 상태점(1)의 바이패스된 공기가 혼합된 상태를 나타낸다.



제 3 절 구성기기 알고리즘 작성

1. 팬

가. 압력독립형

(1) 설명

이 알고리즘은 팬을 통과하는 유량과 입구공기의 상태에 의해 팬동력과 출구공기의 상태를 계산하는 것으로 팬을 통과하는 압력상승이 일정한 경우의 에너지계산에 사용한다.

팬 압력상승이 일정하다는 가정은 일반적으로 많은 표준 HVAC 시스템들에서 급기덕트의 정압이 일정하다고 가정된 것과 동일하다.

이러한 계산의 목적은 팬의 설계가 최대부하일 때의 동력과 정격상태에서 측정된 유량비율에 의해 이루어지기 때문이다. 부분부하에서의 소비동력을 정격유량 대 운전유량 비율의 함수와 같은 경험식의 관계에 의해 결정된다. 경험식에 맞는 계수는 유량제어의 3가지 방법(취출댐퍼제어, 인렛베인제어, 팬속도제어)에 따라 다르게 적용한다.

(2) 알고리즘

HVAC 시스템의 팬 선정은 최대부하 상태에서의 압력상승, 유량, 축동력에 의해서 특성화된다. 팬을 통과함에 따른 압력상승이 정해진 경우에는 정격유량과 축동력의 두 변수가 팬의 정격동력을 결정한다. 정격상태에서의 성능은 정격성능을 이용한 부분부하비율접근법으로 계산한다. 공기유량이 감소될 때의 부분부하 비율은 다음 식 (3.1)로 계산한다.

$$PLR = \frac{m/\rho}{Q_{rat}} \tag{3.1}$$

여기서, PLR : 부분부하시의 풍량 비율 (-)
m : 건공기 질량유량 (kg/s)
ρ : 입구 습공기 밀도 (kg/㎥)
Q_{rat} : 정격체적유량 (㎥/s)

부분부하에서의 팬동력은 정격동력의 비율로 나타낸다.
유량이 감소될 때의 최대부하에 대한 동력의 비율은 다항식 (3.2)를 이용하여 계산한다.

$$FFLP = C_0 + C_1 PLR + C_2 PLR^2 + C_3 PLR^3 \quad (3.2)$$

여기서, $FFLP$: 최대부하일때의 동력에 대한 비율 (-)
C_i : 회귀계수 (-)

부분부하 제어의 3가지 방식에 따라 회귀계수는 다음 표 3.1과 같다.

표 3.1 제어방식에 따른 부분부하 일 때의 회귀계수

제어방식	C_0	C_1	C_2	C_3
취출댐퍼제어	0.3507123	0.3085	- 0.54137	0.871988
입구베인제어	0.3707	0.9725	- 0.3424	0.0
변속운전제어	0.00153	0.005208	1.1086	-0.11635563

부분부하에서의 실제 팬의 축동력은 다음 식 (3.3)으로 계산한다.

$$W_s = FFLP \times W_{s,rat} \quad (3.3)$$

여기서, W_s : 축동력 (W)
$W_{s,rat}$: 정격축동력(W)

모터의 총 동력소비는 다음 식 (3.4)의 모터효율을 이용하여 계산한다.

$$W_t = \frac{W_s}{\eta_{mot}} \quad (3.4)$$

여기서, W_t : 팬 동력 (W)

η_{mot} : 모터의 운전효율

팬을 통과하는 공기의 온도상승은 모터의 위치에 따라 다르다. 만약 모터가 공기흐름의 외부에 위치하면 모터의 비효율에 의한 영향은 열전달에 영향을 주지 않으며, 모터가 공기 흐름의 안쪽에 있다면 전체 동력의 비율에 따라 공기흐름에 열전달이 된다.

일반적인 모델의 모터 열전달은 다음 식 (3.5)로 계산한다.

$$q_{loss} = W_s + (W_t - W_s) \cdot f_{m,loss} \tag{3.5}$$

여기서, $f_{m,loss}$ 는 모터의 비효율이 공기중으로 전달되는 비율이다. 만약 모터가 공기의 흐름 안에 있다면 이 비율은 일정한 값이 되며, 모터가 공기의 흐름밖에 있다면 0이 된다.

출구공기의 상태는 습공기 관계식으로 엔탈피와 절대습도를 이용하여 계산한다.

(3) 프로그램

프로그램의 내용은 부록 B에 있다.

나. 압력종속형

(1) 설명

이 알고리즘은 유량과 입구공기의 상태가 주어지면 팬의 소요동력과 출구공기의 상태를 계산하는 것으로 유량에 따른 압력상승과 효율에 관련된 무차원 성능곡선에 의해 팬의 성능과 출구상태를 결정한다. 이 성능곡선은 사용자에게 경험적 계수의 다항식으로 표현되어 제공된다. 이 알고리즘은 출구공기의 상태를 알고 입구공기의 상태를 계산하려면 음수의 풍량을 입력해야 정확히 계산할 수 있으며, 팬을 통과하는 공기는 밀도가 변하지 않는 것으로 가정한다.

(2) 알고리즘

팬의 성능은 주어진 유량이 장치를 통과함에 따른 압력상승과 소요축동력에 따라 특성화된다. 여기서 이 두 항목은 양정과 효율로 표현되며, 일반적으로 제작자에 의해 유량의 함수로 제공된다. 이러한 성능곡선은 모델에 기초해서 작성되는데 무

차원 변수는 단순화 및 일반화된 모델을 사용한다. 무차원 유량계수와 무차원 양정계수는 다음 식 (3.6)으로 계산한다.

$$\Phi = \frac{m}{\rho N d^3} \qquad \Psi = \frac{\triangle P}{\rho N^2 d^2} \tag{3.6}$$

여기서,

Φ	:	무차원 유량계수(-)
ρ	:	입구공기의 밀도 (kg/㎥)
m	:	건공기 질량유량 (kg/s)
N	:	회전 속도 (hz)
d	:	팬의 직경 (m)
Ψ	:	무차원 양정계수(-)
$\triangle P$	:	압력차 (Pa)

팬의 성능은 제작자의 자료를 이용하여 다음 식 (3.7, 3.8)과 같은 무차원 다항 회귀식으로 계산할 수 있다.

$$\Psi = a_0 + a_1\Phi + a_2\Phi^2 + a_3\Phi^3 + a_4\Phi^4 \tag{3.7}$$

$$\eta_s = b_0 + b_1\Phi + b_2\Phi^2 + b_3\Phi^3 + b_4\Phi^4 \tag{3.8}$$

여기서,

a_i	:	양정의 회귀계수
b_i	:	효율의 회귀계수
η_s	:	팬의 효율

팬의 효율은 다음 식 (3.9)로 계산한다.

$$\eta_s = \frac{m\,\Delta P}{\rho\, W_s} \tag{3.9}$$

식 (3.6)과 식 (3.9)을 이용하면 다음의 식 (3.10)으로 축동력을 구할 수 있다.

$$W_s = \frac{m \Psi N^2 d^2}{\eta_s} \tag{3.10}$$

a_t와 b_t 계수가 주어지면 팬을 통과하면서 일어나는 압력상승은 양정계수로부터 계산한다. 총합시스템동력은 다음 식 (3.11)로 계산한다.

$$W_t = \frac{W_s}{\eta_{mot}} \tag{3.11}$$

여기서, W_t : 팬 동력 (W)

η_{mot} : 모터의 운전효율

팬 모터의 비효율로 인하여 유체의 흐름에 열이 전달된다. 만약 모터가 유체흐름의 밖에 위치하면 일반적으로 0이된다. 그러나 팬에서는 모터가 자주 유체 흐름안에 위치하며 모터의 손실은 바로 공기 온도상승에 영향을 미친다. 다음의 식 (3.12)는 일반적인 모델의 유체에 열전달을 계산하는 식이다.

$$q_{loss} = W_s + (W_t - W_s) \cdot f_{m,loss} \tag{3.12}$$

여기서, q_{loss} : 유체로의 열전달 (W)

$f_{m,loss}$: 유체로 전달되는 모터의 비효율의 비율

출구공기상태는 엔탈피와 절대습도를 이용하여 습공기 식에서 구한다.

(3) 프로그램

프로그램의 내용은 부록 B에 있다.

2. 펌프

가. 압력독립형

(1) 설명

이 알고리즘은 유량과 입구유체의 상태가 주어지면 펌프의 동력과 출구유체의 온도를 계산한다. 이 알고리즘은 상세한 압력조정의 정보가 없을 때 시스템의 에너지 계산에 사용하기 위해 만들었다.

펌프의 소비동력은 부분부하 유량의 기능에 따라 부분부하 소비동력의 회귀식으로부터 계산한다. 이러한 계산의 목적은 펌프의 설계가 정격상태에서의 압력상승, 유량과 펌프 및 모터의 동력소비에 의해 특성화 되기 때문이다.

부분부하에서의 동력소비는 정격유량 대 작동유량의 함수에 의한 경험식으로 결정한다. 펌프를 통과하면서 발생하는 압력상승은 계산하지 않는다.

유체의 온도상승은 샤프트에 입력되는 동력으로 계산되며, 모터효율과 펌프/모터의 총합효율은 일정한 것으로 가정한다. 이 알고리즘은 펌프를 통과하는 유체의 밀도 변화는 무시할만한 것으로 가정한다면 주어진 출구유량상태에서 규정된 부(-)유량율에 의해 입구유량상태를 정확히 계산할 수 있다.

(2) 알고리즘

HVAC 시스템의 펌프선정은 최대부하 상태에서 압력상승, 유량, 동력의 특성에 의한다. 펌프를 통과하는 압력이 주어지면 펌프의 정격에너지 소비는 유량과 동력의 두변수에 의한다. 부분비율 상태의 성능은 부분부하 비율 접근법을 이용하여 정격성능을 계산한다.

유량감소시에 부분부하 비율은 실제 체적유량 대 정격유량의 비로 재조정된다.

$$PLR = \frac{m/\rho}{Q_{rat}} \qquad (3.13)$$

여기서, PLR : 부분부하시의 유량 비율 (-)

m : 유체의 질량유량 (kg/s)

ρ : 입구 유체의 밀도 (kg/㎥)

Q_{rat} : 정격체적유량 (㎥/s)

부분부하의 동력소비는 정격동력의 비율로 나타낼 수 있다.

유량이 감소할 때 최대부하 동력의 비는 부분부하 비율의 기능에 맞는 다항식 곡선에 의한다.

$$FFLP = C_0 + C_1 PLR + C_2 PLR^2 + C_3 PLR^3 \quad (3.14)$$

여기서, $FFLP$: 최대부하일때의 동력에 대한 비율 (-)

C_i : 회귀계수 (-)

회귀계수의 값은 반드시 펌프의 부분부하성능과 시스템의 구성에 의해 결정되어야만 한다. 부분부하에서 실제 펌프/모터의 동력은 최대동력과 최대동력의 비로 계산된다.

$$W_s = FFLP * W_{s,rat} \quad (3.15)$$

여기서, W_s : 축동력 (W)

$W_{s,rat}$: 정격축동력(W)

펌프를 통과하는 유체의 온도상승은 펌프의 비효율에 의해 계산된다. 총 펌프/모터의 효율은 평가된 효율로 일정하다고 가정한다.

$$\eta_t = \frac{Q_{rat} \cdot \Delta P_{rat}}{W_{t,rat}} \quad (3.16)$$

여기서, η_t : 펌프/모터의 총효율

$W_{t,rat}$: 펌프/모터 정격동력 (W)

ΔP_{rat} : 정격 펌프 양정(Pa)

Q_{rat} : 정격체적유량(㎥/s)

펌프의 효율은 총효율과 모터효율로부터 계산할 수 있다.

$$\eta_p = \frac{\eta_t}{\eta_m} \quad (3.17)$$

단열 펌핑(Pumping)으로 가정하면 유체의 온도상승은 다음 식 (3.18)과 같이 총 입력 동력과 관계가 있다.

$$f_{m,loss}(1-\eta_m)W_t+(1-\eta_p)\eta_m W_t = mC_p\Delta T \qquad (3.18)$$

위쪽의 첫째항은 유체의 온도상승에 의한 모터의 비효율의 효과를 계산한 것이고 두 번째 항은 펌프의 비효율의 효과를 계산한 것이다. 변수 $f_{m,loss}$ 는 모터 손실의 비율이며 이것은 유체의 흐름에 따라 변동한다. 펌프에서는 이 값이 일반적으로 0 이거나 매우 작다.

나. 압력종속형

(1) 설명

이 알고리즘은 유량과 입구유체의 상태가 주어지면 펌프의 소요동력과 출구유체의 상태를 계산하는 것으로 유량에 따른 압력상승과 효율에 관련된 무차원 성능곡선에 의해 펌프의 성능과 출구상태를 결정한다. 이 성능곡선은 사용자에게 경험적 계수의 다항식으로 표현되어 제공된다. 이 알고리즘은 출구공기의 상태를 알고 입구공기의 상태를 계산하려면 음수의 풍량을 입력하여 계산할 수 있으며, 팬을 통과하는 공기는 밀도가 변하지 않는 것으로 가정한다.

(2) 알고리즘

펌프의 성능은 주어진 유량이 장치를 통과함에 따른 압력상승과 소요축동력에 따라 특성화된다. 여기서 이 두 항목은 양정과 효율로 표현되며, 일반적으로 제작자에 의해 유량의 함수로 제공된다. 이러한 성능곡선은 모델에 기초해서 작성되는데 무차원 변수는 단순화 및 일반화된 모델을 사용한다. 무차원 유량계수와 무차원 양정계수는 다음 식 (3.19)로 계산한다.

$$\Phi = \frac{m}{\rho N d^3} \qquad \Psi = \frac{\Delta P}{\rho N^2 d^2} \qquad (3.19)$$

여기서, Φ : 무차원 유량계수(-)
ρ : 입구유체의 밀도 (kg/㎥)
m : 유체의 질량유량 (kg/s)
N : 회전 속도 (hz)
d : 펌프의 직경 (m)
Ψ : 무차원 양정계수(-)
ΔP : 압력차 (Pa)

펌프의 성능은 제작자의 자료를 이용하여 다음 식 (3.20, 3.21)과 같은 무차원 다항회귀식으로 계산할 수 있다.

$$\Psi = a_0 + a_1\Phi + a_2\Phi^2 + a_3\Phi^3 + a_4\Phi^4 \tag{3.20}$$

$$\eta_s = b_0 + b_1\Phi + b_2\Phi^2 + b_3\Phi^3 + b_4\Phi^4 \tag{3.21}$$

여기서, a_i : 양정의 회귀계수
b_i : 효율의 회귀계수
η_s : 펌프의 효율

펌프의 효율은 다음 식 (3.22)로 계산한다.

$$\eta_s = \frac{m\ \Delta P}{\rho\ W_s} \tag{3.22}$$

식 (3.19)와 식 (3.22)를 이용하면 다음의 식 (3.23)으로 축동력을 구할 수 있다.

$$W_s = \frac{m\ \Psi N^2 d^2}{\eta_s} \tag{3.23}$$

a_i와 b_i 계수가 주어지면 펌프을 통과하면서 일어나는 압력상승은 양정계수로부터 계산한다. 총합시스템동력은 다음 식 (3.24)로 계산한다.

$$W_t = \frac{W_s}{\eta_{mot}} \tag{3.24}$$

여기서, W_t : 펌프 동력 (W)
η_{mot} : 모터의 운전효율

펌프 모터의 비효율로 인하여 유체의 흐름에 열이 전달된다. 만약 모터가 유체흐름의 밖에 위치하면 일반적으로 0이된다. 다음의 식 (3.25)는 일반적인 모델의 유체에 열전달을 계산하는 식이다.

$$q_{loss} = W_s + (W_t - W_s) \cdot f_{m,loss} \tag{3.25}$$

여기서, q_{loss} : 유체로의 열전달 (W)
$f_{m,loss}$: 유체로 전달되는 모터의 비효율의 비율

출구유체상태는 엔탈피와 절대습도를 이용하여 습공기 식에서 구한다.

3. 밸브

가. 밸브와 배관

(1) 설명

이 알고리즘은 연속된 밸브 또는 댐퍼(변유량 저항)을 통과하고 일정한 유체저항이 있는 회로의 유량을 계산하기 위한 것이다. 유량은 밸브어사러티(valve authority)로 정의된 밸브의 위치를 이용한 함수를 이용하여 결정한다. 주어진 밸브어사러티와 밸브가 모두 열렸을 때의 유량특성, 유량은 밸브 위치의 함수를 이용하여 결정할 수 있다. 단 시스템상의 상세한 압력조정의 실행은 무시한다. 이 알고리즘을 적용하기 위한 중요한 제한사항으로 연결된 구성품을 통과하는 총압력강하는 밸브의 위치변화에 따라 반드시 일정해야 한다. 이것은 특히 큰 회로의 출발점에서 보다 정확하다.

(2) 알고리즘

밸브는 유량에 따라 저항이 변동할 수 있도록 모델화 할 수 있다. 밸브를 통과함에 따른 압력강하는 유량저항계수와 유량으로 다음 식 (3.26)으로 계산한다.

$$\triangle P = K\, m^2 \tag{3.26}$$

여기서, $\triangle P$: 밸브를 통과하는 압력차이(Pa)
K : 유량저항계수
m : 유량(kg/s)

저항은 밸브가 전개 할 때와 유량에 따라 축의 움직임이 변함에 따라 어떤 공칭값을 가지고 있다. 밸브를 전개(全開)할 때의 저항 Ko는 밸브선정의 변수가 된다. 밸브의 위치에 따른 저항의 변화는 밸브의 물리적인 구조에 의해 결정된다.(예, 플러그와 시트의 모양) 밸브를 통과하는 압력강하가 일정하면 저항의 변화는 유량의 변화에 따른다. 압력강하가 일정할 때 밸브위치에 따른 유량의 변화는 밸브의 특성을 측정하는 가장 일반적인 방법이다. 예를 들면 어떤 유량의 밸브는 밸브 축의 위치에 직접 비례하는 선형밸브로 정의된다.

유량저항계수 K는 넓은 범위의 유량특성을 선형과 지수밸브 저항의 가중평균으로 정의하여 설명할 수 있다.

$$K = \frac{W_f K_0}{[(1-\lambda)C+\lambda]^2} + (1 - W_f)K_0\,\lambda^{(2C-2)} \tag{3.27}$$

여기서, W_f : 밸브 특성에 따른 가중계수
W = 0 : 지수형 밸브
W = 1 : 선형 밸브
0<W<1 : 중간형 밸브
K_0 : 밸브 전개시의 유량저항계수
m : 유량(kg/s)
λ : 누설변수
(일정한 압력차에서 전폐시의 유량흐름과 전개시의 유량흐름의 비)
C : 밸브의 위치

첫째항은 선형밸브의 저항이고 두번째항은 지수밸브의 저항이다. W_f는 상술한 두 항의 유량특성을 조합한 가중요소이다. 가중계수의 0은 지수밸브, 가중계수 1은 선형밸브, 그리고 가중계수의 중간값은 밸브의 형태가 중간형인 것을 나타낸다. 밸브어사러티는 밸브를 전개 했을 때 밸브를 통과하는 압력강하와 전시스템의 압력강하와의 비율로 정의된다.

$$a = \frac{\Delta P_{valve,o}}{\Delta P_{tot}} \qquad (3.28)$$

여기서,
a : 밸브어사러티
$\Delta P_{valve,o}$: 밸브의 통과에 따른 압력강하(Pa)
ΔP_{tot} : 전 시스템의 압력강하(Pa)

밸브어사러티는 제어밸브 제작의 설계기준으로 사용된다. 이것은 또한 전개상태에서 설계조건에 의해 발생하는 밸브와 시스템을 통과하는 압력강하와 설계유량의 지식으로부터 얻을 수 있다. 만약 밸브어사러티가 압력강하가 고정된 유량저항으로 가정한다면 밸브어사러티는 상사저항계수 K_{fixed}를 통하여 유량의 제곱에 비례한다. 연결된 시스템의 총압력강하는 다음 식으로 계산할 수 있다.

$$\Delta P_{tot} = \Delta P_{valve} + \Delta P_{fixed} = (K + K_{fixed})\, m \qquad (3.29)$$

밸브어사러티는 두 개의 유량저항 K_o, K_{fixed} 으로 정의할 수 있다.

$$a = \frac{K_o}{K_o + K_{fixed}} \qquad (3.30)$$

만약 총압력강하가 밸브의 위치에 의존한다면 밸브의 위치가 정해지지 않은 경우의 유량은 전개시의 유량과 저항계수의 함수와의 관계에 의해서 구할 수 있다.

$$\frac{m}{m_o} = \sqrt{\frac{K_o + K_{fixed}}{K + K_{fixed}}} \qquad (3.31)$$

여기서, m_o : 밸브 전개시의 설계유량(kg/s)

밸브어사러티의 정의를 이용하면 위의 식은 주어진 시스템의 압력에 대한 항을 없애고 유량의 비율에 관한 식으로 고칠 수 있다.

$$m = m_o \sqrt{\frac{1}{a\frac{K}{K_o} + (1-a)}} \qquad (3.32)$$

제 4 절 유틸리티 알고리즘 작성 및 S/W 개발

1. 벽의 열취득

가. 열관류율

(1) 설 명

벽체를 통한 열전달율(Q)은 다음과 같다.

$$Q = U A \Delta T \qquad (4.1)$$

여기서, ΔT는 정상열전달계산에서는 실내와 외기의 온도차이며, 주기정상열전달계산법인 ASHRAE 냉방부하계산에서는 냉방부하온도차, CLTD 이고 일본공기조화·위생공학회 계산법에서는 효과온도차, ETD 이다.

(2) 알고리즘

벽체의 열관류율 U는 다음과 같다.

$$U = \frac{1}{R_T} = \frac{1}{\frac{1}{h_o} + \Sigma \frac{\ell_i}{k_i} + \frac{1}{C} + \frac{1}{h_i}} \qquad (4.2)$$

여기서, U : 벽체의 열관류율($W/m^2 K$)
R_T : 열관류저항($m^2 K / W$)
h_o : 외표면열전달계수($W/m^2 K$)
h_i : 벽체의 내표면열전달계수($W/m^2 K$),
ℓ : 각 재료의 두께(m)
k : 각 재료의 열전도율($W/m K$)
C : 중공층이 있는 경우 중공층의 콘덕턴스($W/m^2 K$)

(3) 프로그램

부록 C는 열관류율, 응답계수 및 전도전달함수를 구하는 프로그램이며, menu=1로

두어 열관류율을 계산한다.

(가) 입력화면

그림 4.1은 입력화면의 일예이며, 열관류율을 계산하기 위한 6가지의 입력데이터의 입력방법은 다음과 같다.

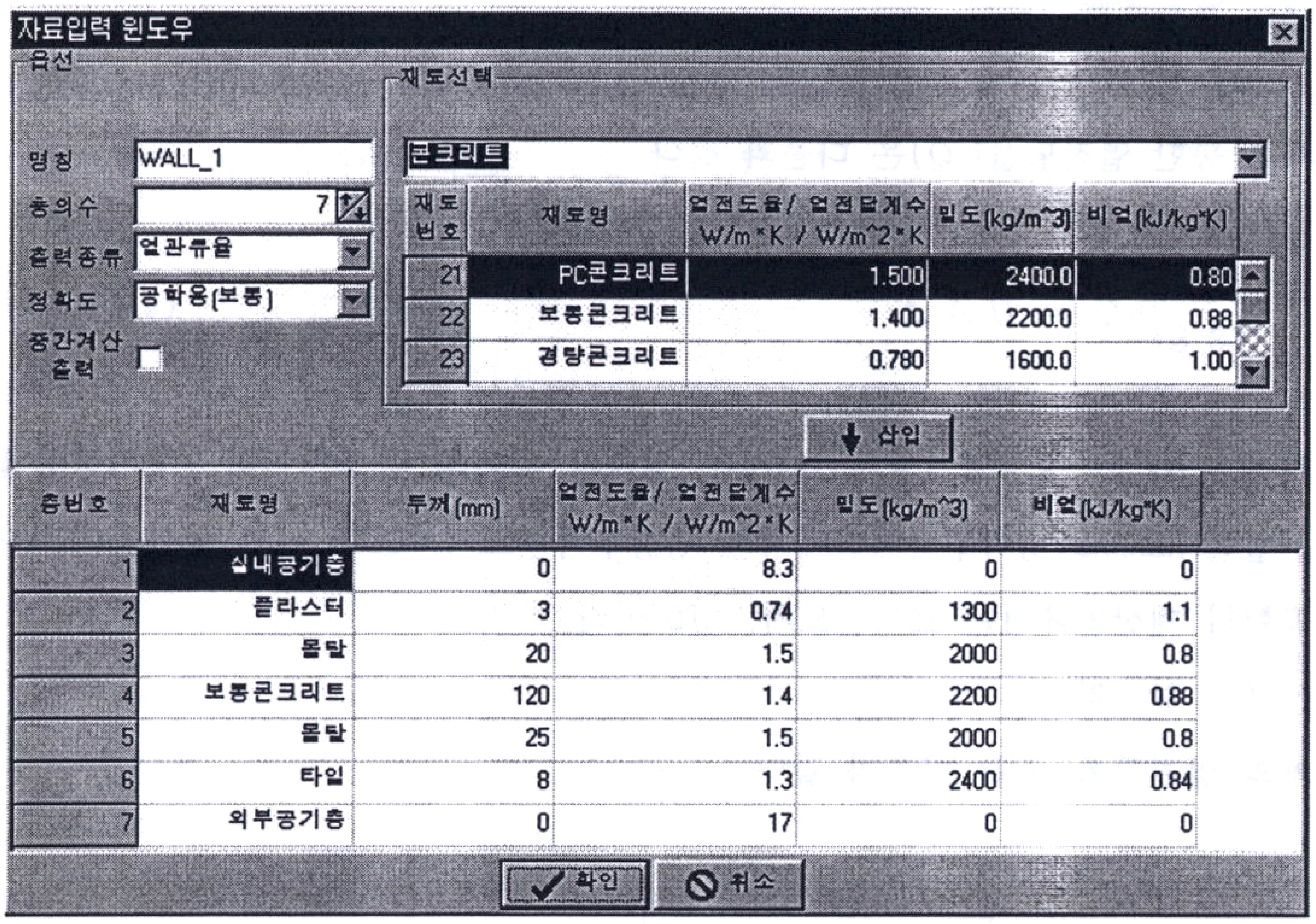


그림 4.1 열관류율 입력화면

1. 명칭 - 명칭은 한글 32자, 영문은 64자 이내의 문자열을 입력할 수 있다.

2. 층의 수 - 층의 수는 실내공기층과 외부공기층을 포함한 층의 개수이며, 최대 14개 이다.

3. 출력종류 - 출력종류는 열관류율, 응답계수, 전도전달함수의 3가지 경우가 있으며 기본출력은 열관류율로 설정되어 있다.

4. 정확도 - 정확도는 연구용, 공학용(보통), 공학용(정밀)의 3가지 경우가 있으며 공학용(보통)이 기본값으로 설정되어 있다.

5. 중간계산출력 - 중간계산 출력은 없다.

6. 재료데이터 입력방법

가. 재료는 실내 층에서 실외측의 순서로 입력한다.

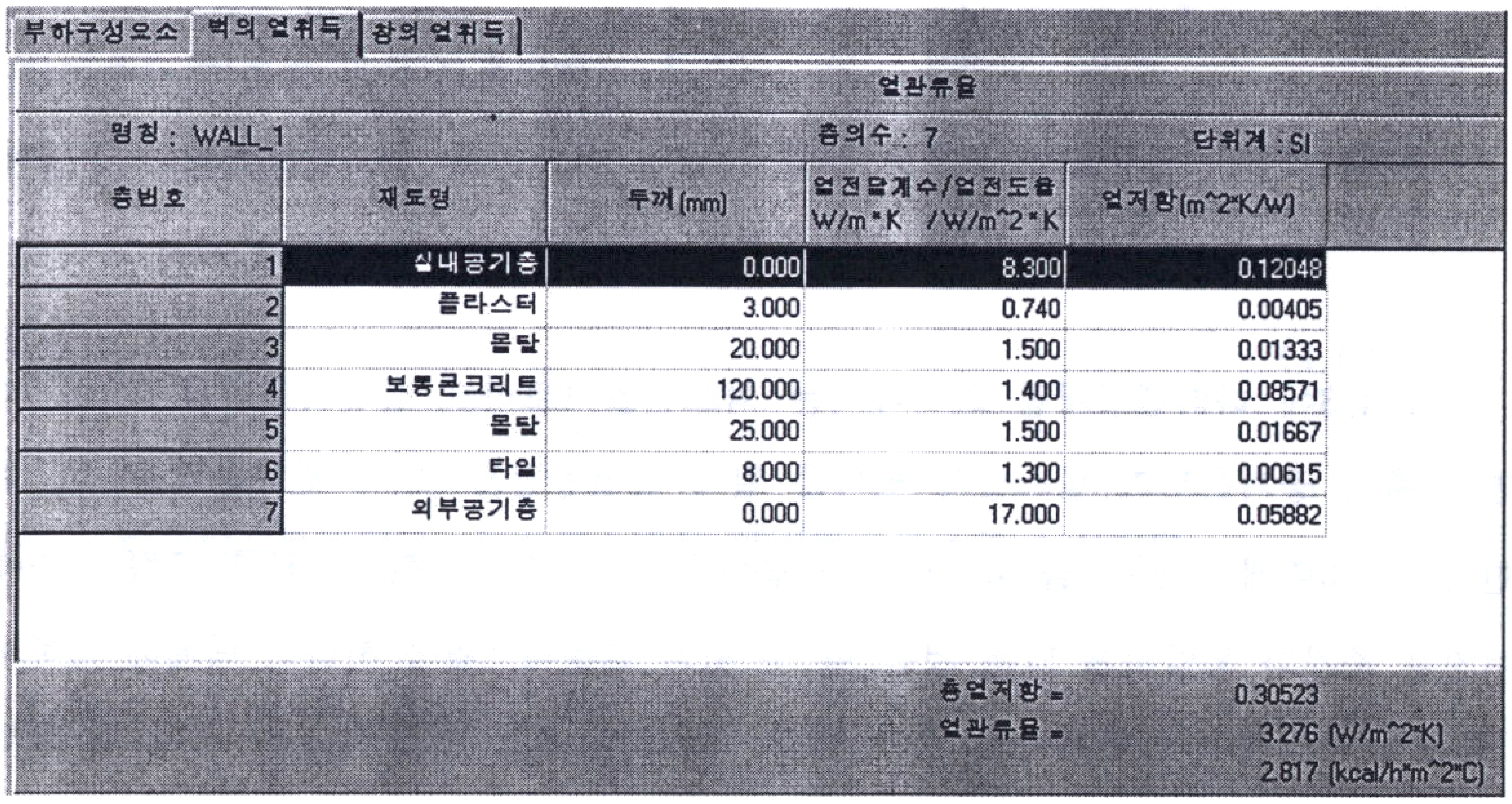
나. 각 층의 재료데이터는 재료선택창에서 재료명에 해당하는 재료번호를 선택하고 “삽입” 버튼을 선택하면 열전도율, 밀도 및 비열이 자동적으로 입력되므로 두께만 입력하면 된다. 공기층인 경우에는 그 두께를 반드시 0으로 입력해야 한다.

다. 해당재료명이 재료표에 없는 경우에는 등록한 후 ‘나’항과 같이 입력한다

라. 단위계는 SI단위계를 사용한다.

(나) 출력화면

그림 4.2는 출력화면으로 출력값은 각 층의 열저항, 총열저항 및 열관류율이다.



층번호	재료명	두께(mm)	열전달계수/열전도율 W/m*K /W/m^2*K	열저항(m^2*K/W)
1	실내공기층	0.000	8.300	0.12048
2	플라스터	3.000	0.740	0.00405
3	몰탈	20.000	1.500	0.01333
4	보통콘크리트	120.000	1.400	0.08571
5	몰탈	25.000	1.500	0.01667
6	타일	8.000	1.300	0.00615
7	외부공기층	0.000	17.000	0.05882

그림 4.2 열관류율의 출력화면

나. 응답계수(Response Factor)

(1) 설명

벽체의 응답계수로는 그림 4.3에 나타낸 것과 같이 0℃ 벽체의 한 면에서 여진으로 삼각파 펄스상의 온도변화를 줄 때, 각 시간간격 Δt 의 표면열류 응답을 나타내는 수열이다. 응답은 여진을 받은 표면에서의 것과 여진의 반대측 표면에서의 것 2가지가 있는데, 전자를 흡열응답, 후자를 관류열응답이라 한다. 온도여진의 위치

는 경계조건에 따라서 표면온도의 경우와 공기온도의 경우가 있지만, 여기서는 단지 공기온도여진으로 취급한다.

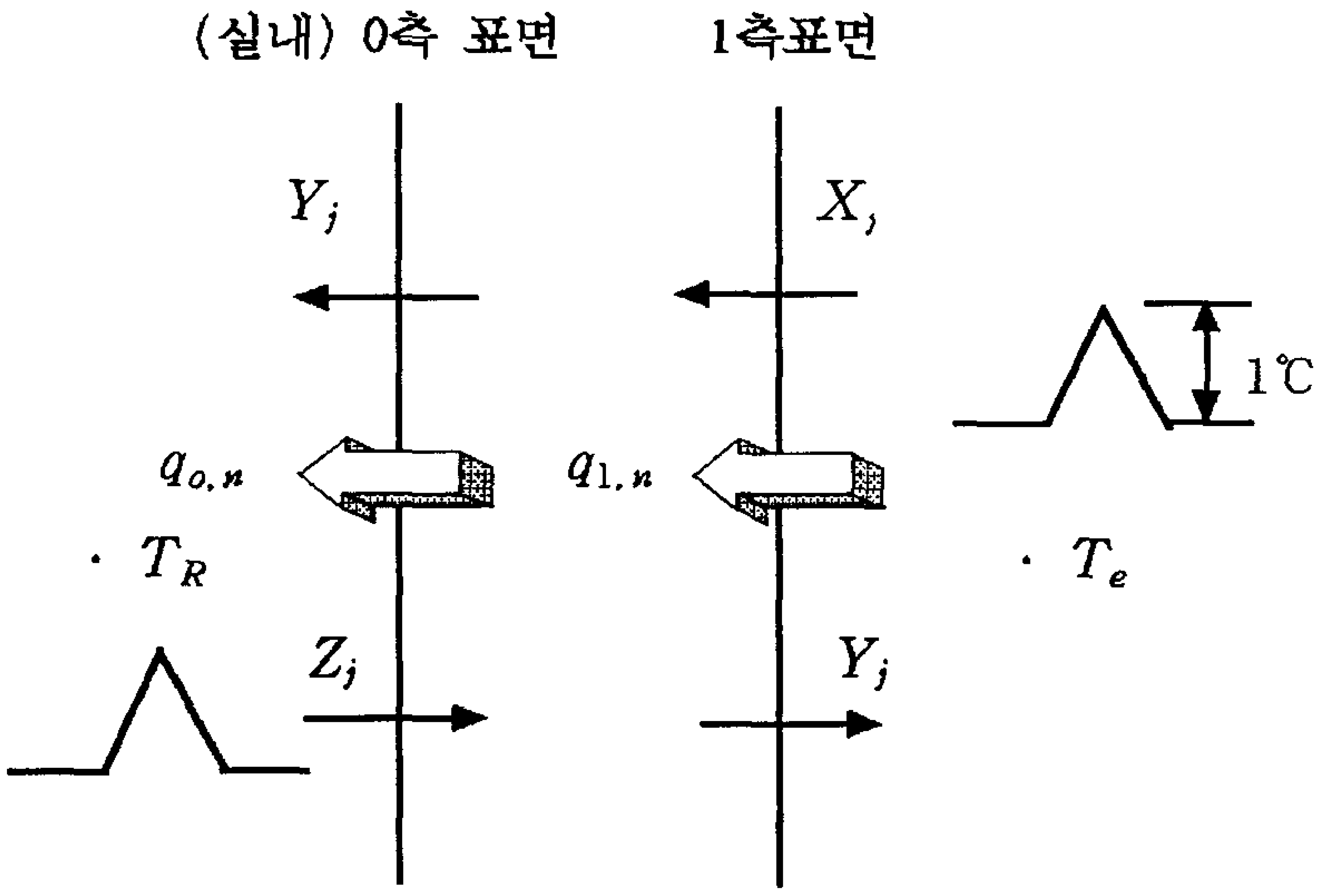


그림 4.3 벽체의 응답계수

그림 4.3에 나타낸것처럼 외벽에서는 외표면을 1로하고 실내측표면을 0으로 하면, 1측의 공기온도가 펄스상의 변화를 할때 흡열응답을 X_j , 관류응답을 Y_j 로 한다. 또 0측에 여진을 줄 때, 이때의 흡열응답을 Z_j 로 한다. 관류응답은 1측에 여진을 준것과 동일하게 되는 것을 알 수 있기 때문에, 결국 하나의 벽체에 대해서 3종류의 응답계수가 존재하는 것이 된다. X_j, Y_j, Z_j는 여진후 $j\Delta t$ 시의 각 표면에서의 열류이고 단위는 W/㎡ 이다.

실용계산에서는 응답계수의 감소율인 공비 C를 이용한 수정응답계수 XM_j, YM_j, ZM_j 를 사용하며 다음 식으로 구한다.

$$XM_j = X_j - C * X_{j-1}, \qquad 0 < j < N \qquad (4.3)$$

여기서, N는 개별 계산항의 수 이다.

열전달율은 표면 1에서 표면 0방향을 +로하면 그림 4.3에서 n시의 열전달 $q_{1,n}$과

$q_{0,n}$는 다음과 같다.

$$q_{1,n} = \sum_{j=0}^{\infty} X_j \cdot T_{e,n-j} - \sum_{j=0}^{\infty} Y_j \cdot T_{R,n-j} \tag{4.4}$$

$$q_{0,n} = \sum_{j=0}^{\infty} Y_j \cdot T_{e,n-j} - \sum_{j=0}^{\infty} Z_j \cdot T_{R,n-j}$$

수정응답계수를 이용하면 다음과 같다.

$$q_{0,n} = \sum_{n=0}^{N} YM_j \cdot T_{e,n-j} - \sum_{j=0}^{N} ZM_j \cdot T_{R,n-j} + c\, q_{o,n-1} \tag{4.5}$$

응답계수는 다음과 같이 도출된다.

두께 l_i , 열전도율 k_i, 열확산율 a_i인 i 층의 열전도 방정식은 다음과 같다

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{a} \frac{\partial T}{\partial t} \tag{4.6}$$

이것을 라프라스변환하고, 라프라스변환 매개변수를 s로 표시하면,

$$\begin{bmatrix} T_i(s) \\ q_i(s) \end{bmatrix} = \begin{bmatrix} A_i(s) & B_i(s) \\ C_i(s) & D_i(s) \end{bmatrix} \begin{bmatrix} T_{i+1}(s) \\ q_{i+1}(s) \end{bmatrix} \tag{4.7}$$

여기서,

$$A_i(s) = \cosh l_i \sqrt{s/a_i}$$

$$B_i(s) = \frac{1}{k_i \sqrt{s/a_i}} \cdot \sinh l_i \sqrt{s/a_i} \tag{4.8}$$

$$C_i(s) = k_i \sqrt{s/a_i} \cdot \sinh l_i \sqrt{s/a_i}$$

$$D_i(s) = A_i(s)$$

중공층이나 공기층 같이 열용량이 없는 경우에는 다음과 같다.

$$A_i(s) = D_i(s) = 1$$

$$B_i(s) = R_i = \frac{l_i}{k_i} \tag{4.9}$$

$$C_i(s) = 0$$

n개의 다층벽과 실내와 실외의 공기층을 포함하면 n+2개의 특성행렬로 표시할 수 있다.

$$\begin{bmatrix} T_o(s) \\ q_o(s) \end{bmatrix} = \begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix} \begin{bmatrix} T_R(s) \\ q_R(s) \end{bmatrix} \tag{4.10}$$

우변의 열전달특성 행렬은 다음과 같다.

$$\begin{bmatrix} A(s) & B(s) \\ C(s) & D(s) \end{bmatrix} = \begin{bmatrix} 1 & R_o \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A_1(s) & B_1(s) \\ C_1(s) & D_1(s) \end{bmatrix} \cdots \begin{bmatrix} A_n(s) & B_n(s) \\ C_n(s) & D_n(s) \end{bmatrix} \begin{bmatrix} 1 & R_R \\ 0 & 1 \end{bmatrix} \tag{4.11}$$

열량으로 정리하면

$$\begin{bmatrix} q_o(s) \\ q_R(s) \end{bmatrix} = \begin{bmatrix} \frac{D(s)}{B(s)} & -\frac{1}{B(s)} \\ \frac{1}{B(s)} & -\frac{A(s)}{B(s)} \end{bmatrix} \begin{bmatrix} T_o(s) \\ T_R(s) \end{bmatrix} \tag{4.12}$$

단위응답을 구하기 위해서 $T_o(t) = 1$, $T_R(t) = 0$ 혹은 $T_o(t) = 0$, $T_R(t) = 1$ 로 두고, 이를 라프라스 변환하여 식 (4.12)에 대입하면 라프라스 변환된 열류응답의 일반적인 표시는

$$q(s) = \frac{F(s)}{sB(s)} \tag{4.13}$$

가 되며, 여기서 F(s)는 특성행열요소중의 A(s), D(s) 혹은 1중의 하나가 된다. 식 (4.13)의 분모에서

$$B(s)=0 \tag{4.14}$$

가 특성방정식이며, 그 근을 $\beta_k(k=1, 2, \cdot\cdot\cdot)$라 하면 단위열류응답은 식 (4.13)을 유수정리에 의해 역변환하여 구할 수 있다.

$$q(t)=\lim_{s\to 0}\frac{d}{ds}\left(\frac{sF(s)e^{st}}{B(s)}\right)+\sum_{k=1}^{\infty}\left(\frac{F(s)e^{st}}{s\cdot\frac{dB(s)}{ds}}\right)_{s=-\beta_k} \tag{4.15}$$

식 (4.15)의 우변 첫항은 극(pole) 영에서의 유수이며, s=0일 때 행열요소 F(s)는 항상 1이므로 1/B(s)이 된다.
그런데 식 (4.11)로부터

$$\lim_{s\to 0}\begin{pmatrix} A(s) & B(s) \\ C(s) & D(s) \end{pmatrix}=\begin{pmatrix} 1 & \frac{1}{U} \\ 0 & 1 \end{pmatrix} \tag{4.16}$$

이 되므로 U = 1 / B(o)이며 다층벽의 열관류율을 뜻한다. 그러므로 단위응답의 기본해는 식 (4.15)로 부터 다음의 급수해로 표시된다.

$$q(t)=A_o+\sum_{k=1}^{\infty}A_k e^{-\beta_k\cdot t} \tag{4.17}$$

여기서,

$$A_o=U \tag{4.18}$$

$$A_k=\left\{\frac{F(s)}{s\cdot dB(s)/ds}\right\}_{s=-\beta_k}$$

RF의 종류는 식 (4.18)에서 $F(s)=D(s)$ 일 때 X계열, $F(s)=1$ 일 때 Y계열 그리고 $F(s)=A(s)$ 일 때 Z계열이 되며, 그림 4.1과 같다.
여기서, 열전달행렬 요소의 값은 표 4.1과 같다.

표 4.1 특성행열 요소의 값

구분	행열요소	행열요소의 미분값	
		행열요소 S=0일때	$S=-\beta_k(k>1)$일때
고체층	$A_i(-\beta_k)=\cos w$	$A_i'(-\beta_k)=\frac{R_i C_i}{2}$	$\frac{R_i C_i}{2}\frac{\sin w}{w}$
	$B_i(-\beta_k)=\frac{R_i}{w}\cdot\sin w$	$B_i'(-\beta_k)=\frac{R_i^2 C_i}{6}$	$\frac{R_i}{2\beta_k}(\frac{\sin w}{w}-\cos w)$
	$C_i(-\beta_k)=-\frac{w}{R_i}\sin w$	$C_i'(-\beta_k)=C_i$	$\frac{C_i}{2}(\frac{\sin w}{w}+\cos w)$
	$D_i(-\beta_k)=\cos w$	$D_i'(-\beta_k)=\frac{R_i C_i}{2}$	$=A_i'$
공기층	$A_i(0)=1$	0	0
	$B_i(0)=R_i$	0	0
	$C_i(0)=0$	0	0
	$D_i(0)=1$	0	0

주 : $R_i=\frac{\ell_i}{k_i}$, $C_i=\rho_i c_i \ell_i$, $w=\sqrt{R_i C_i \beta_k}$

식 (4.18)의 단위응답은 기본해로서 계가 영도로 유지되어 오다가 t = 0에서 급격히 계단형으로 1℃ 상승하여 그후 일정하게 유지되는 조건에서 구한 것이다. 그러나 외란이 기온인 경우에는 2등변삼각형, 조명인 경우에는 직사각형으로 모사 되므로 Duhamel의 중첩적분식으로 계산하면 다음의 표 4.2와 같다.

표 4.2 RF의 계산식

단위파형	RF_j (j ; 시각열)	
	j = 0	j ≥ 1
2등변삼각파	$A_o+\frac{1}{\delta}\sum_{k=1}^{\infty}\frac{A_k}{\beta_k}(1-e^{-\beta_k\delta})$	$-\frac{1}{\delta}\sum_{k=1}^{\infty}\frac{A_k}{\beta_k}(1-e^{-\beta_k\delta})^2 e^{-(j-1)\beta_k\delta}$
직사각형파	$A_o+\sum_{k=1}^{\infty}A_k$	$\sum_{k=1}^{\infty}A_k(1-e^{\beta_k\delta})e^{-j\beta_k\delta}$
직각삼각파A	$\frac{1}{\delta}\sum_{k=1}^{\infty}\frac{A_k}{\beta_k}(1-e^{-\beta_k\delta}-\beta_k\cdot\delta)$	$\frac{1}{\delta}\sum_{k=1}^{\infty}\frac{A_k}{\beta_k}(1-e^{-\beta_k\delta}-\beta_k\delta)e^{-j\beta_k\delta}$
직각삼각파B	$A_o+\sum_{k=1}^{\infty}A_k$	$\frac{1}{\delta}\sum_{k=1}^{\infty}\frac{A_k}{\beta_k}(1-e^{\beta_k\delta}+\beta_k\delta)e^{-j\beta_k\delta}$

주 : δ는 시간길이이며, 보통 δ=1이고, $A_o=U$ 이다.

(2) 알고리즘

(가) 준비계산

1. 특성방정식의 최대 근 β_{max} 및 최대항수 N_t 설정

2. 각층의 열저항 R_i와 열용량 C_i 계산

$$R_i=\frac{\ell_i}{k_i}, \qquad C_i=\rho_i c_i \ell_i$$

3. 열관류율 U 계산

(나) 특성방정식의 근 및 예비계산

1. 초기치 $\beta_k=0$ 으로 설정

2. 행열요소 및 그 미분값 계산

· 표 4.1의 식을 이용하여 공기층, 고체층, $\beta_k=0$ 및 $\beta_k\neq0$를 구분하여 계산

3. 관류 및 흡열 응답계수를 구하는 시계열의 계수 계산

· 식 (4.18)과 표 4.1의 식

4. 근 β_k 의 계산

· Newton-Raphson 법에 의한 다음 식에서 새로 구한 현재값 β_k'

가 종전값 β_k에 수렴될 때 까지 반복계산

$$\beta_k' = \beta_k + \frac{B(-\beta_k)}{B'(-\beta_k) - B(-\beta_k)\sum_{j=1}^{m-1}[1/(\beta_j - \beta_k)]} \quad , \ m \geq 2 \qquad (4.19)$$

5. 관류계수 및 흡열계수 계산

· 식 (4.18) 및 표 4.1 이용

(다) 응답계수의 계산

· 표 4.2의 계산식

(라) 공통비와 수정응답계수의 계산

· 공통비는 $c = \exp(-\beta_1)$ 로 계산

· 수정응답계수는 식 (4.3)으로 계산

(3) 프로그램

부록 C에서 menu=2로 두어 응답계수를 계산한다.

(가) 입력화면

입력화면은 그림 4.1의 열관류율 입력화면과 같으나, 출력종류를 응답계수로 설정하면 된다. 중간계산출력이 필요하면 클릭한다.

(나) 출력화면

1. 중간계산값

그림 4.4는 중간계산 출력화면으로 입력 및 중간계산값, 특성방정식 B(S)=0의 근과 응답계수 Y계열과 Z계열의 계산에 필요한 계수들의 값이다.

1. 입력 및 중간계산값

층 번호	재료명	두께 (mm)	k or h (W/m*K) (W/m^2*K)	밀도 (kg/m^3)	비열 (kJ/kg*K)	열저항 (m^2*K/W)	열용량 (W*h/m^2*K)
		입력값				계산값	
1	실내공기층	0.000	8.300	0.000	0.000	0.12048	0.00000
2	플라스터	3.000	0.740	1300.000	1.100	0.00405	1.19167
3	몰탈	20.000	1.500	2000.000	0.800	0.01333	8.88889
4	보통콘크리트	120.000	1.400	2200.000	0.880	0.08571	64.53334
5	몰탈	25.000	1.500	2000.000	0.800	0.01667	11.11111
6	타일	8.000	1.300	2400.000	0.840	0.00615	4.48000
7	외부공기층	0.000	17.000	0.000	0.000	0.05882	0.00000
				합계 =		0.30523	90.20501
				열관류율 =		3.27624(W*h/m^2*K)	

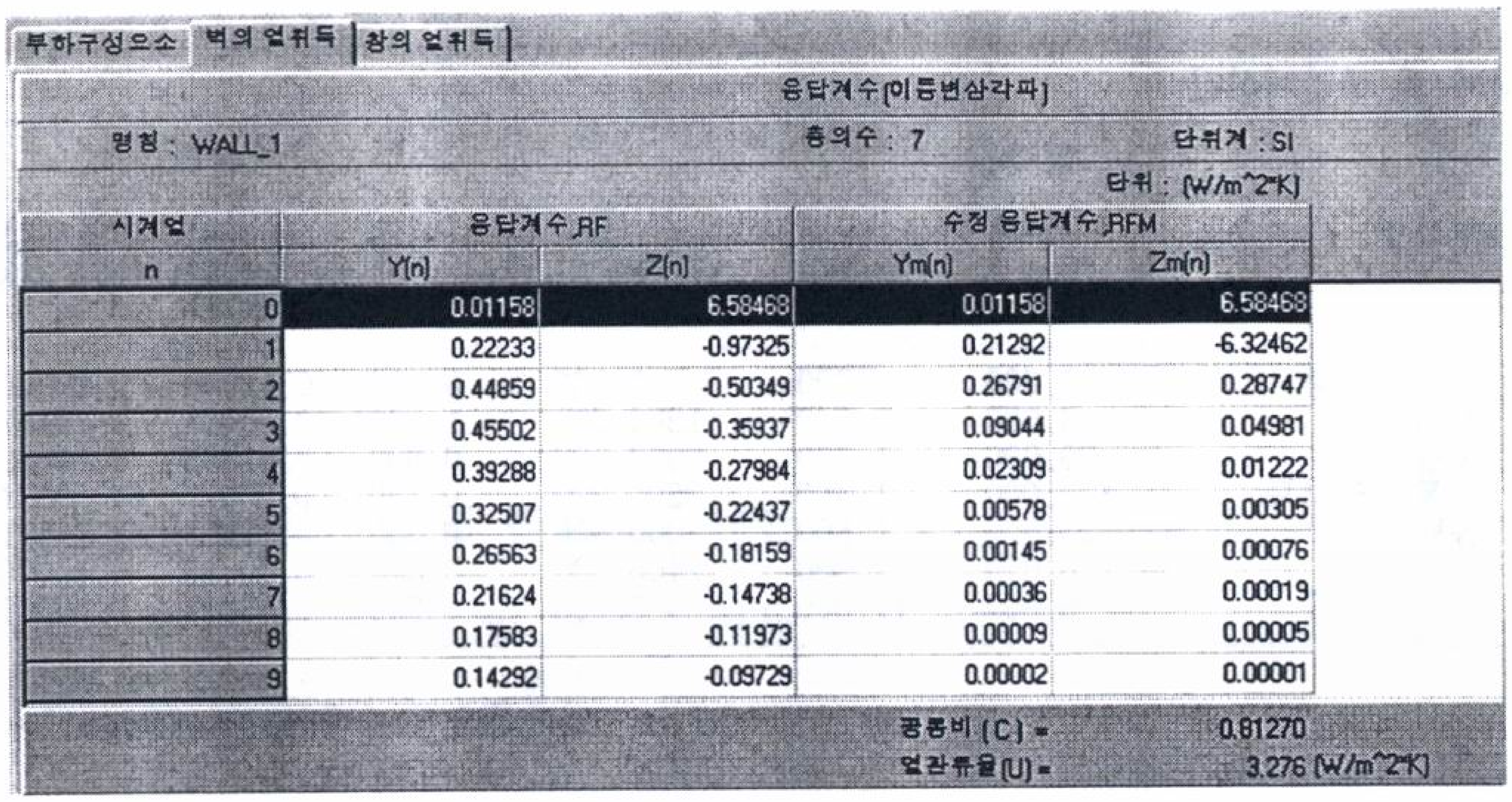
2.특성방정식 B(S)=0의 근 및 계수

근번호	근 BETA(i)	계수 관류계수(A(i,2))	흡열계수(A(i,3))
0	0.20739	-4.44011	3.02229
1	1.38624	1.61994	0.85516
2	4.14253	-0.68720	0.33691
3	8.55119	0.36606	0.17806
4	14.64481	-0.21983	0.11137
5	22.53523	0.14207	0.07537
6	32.28883	-0.09862	0.05286
7	43.85395	0.07347	0.03837
8	57.12386	-0.05762	0.02941
9	72.09834	0.04638	0.02420
.	.	.	.
45	1765.99655	0.00185	0.00099
46	1846.24056	-0.00174	0.00091

그림 4.4 응답계수의 중간계산값

2. 최종결과

그림 4.5는 최종결과의 출력화면으로 응답계수, 수정응답계수 및 공통비의 값이 표시되어 있다.



부하구성요소 | 벽의 열취득 | 창의 열취득

응답계수(이등변삼각파)

명칭 : WALL_1　　층의수 : 7　　단위계 : SI

단위 : (W/m^2*K)

시계열	응답계수 RF		수정 응답계수 RFM	
n	Y(n)	Z(n)	Ym(n)	Zm(n)
0	0.01158	6.58468	0.01158	6.58468
1	0.22233	-0.97325	0.21292	-6.32462
2	0.44859	-0.50349	0.26791	0.28747
3	0.45502	-0.35937	0.09044	0.04981
4	0.39288	-0.27984	0.02309	0.01222
5	0.32507	-0.22437	0.00578	0.00305
6	0.26563	-0.18159	0.00145	0.00076
7	0.21624	-0.14738	0.00036	0.00019
8	0.17583	-0.11973	0.00009	0.00005
9	0.14292	-0.09729	0.00002	0.00001

공통비 (C) = 0.81270

열관류율(U) = 3.276 (W/m^2*K)

그림 4.5 응답계수의 출력화면

다. 전도전달함수(Conduction transfer function)

(1) 설명

전도전달함수에 의한 외벽 및 지붕을 통한 전달은 다음식으로 계산한다.

$$q_n = A\left[\sum_{i=0} b_i (t_{e,n-i}) - \sum_{i=1} d_i \{ (q_{e,n-i})/A \} - t_r \sum_{i=0} c_i \right] \quad (4.20)$$

여기서,

q_n : n시에서의 벽 또는 지붕을 통한 실내로 열전달(W)

A : 벽, 지붕의 실내표면적

n : 시각

i : 합산을 위한 지수

$t_{e,n-i}$: 시간 $n-i$ 일 때의 상당외기온도(℃)

t_r : 일정하다고 가정한 실내온도(℃)

b_i, c_i, d_i : 전도전달함수의 계수들

전도전달함수의 산출에서 실온변동은 연속계이나 연산의 편의상 편의상 1시간 간격의 램프(ramp) 값으로 샘플링 할 수 있다.

$$f*(t) = \sum_{i=0}^{\infty} f(i\triangle)\,\delta(t - i\triangle)$$

라프라스 변환을 하면

$$L[f*(t)] = f*(s) = \sum_{i=0}^{\infty} f(i\triangle)\,e^{-i\triangle s}$$

Z변환은 변수 $Z = e^{\triangle s}$ 로 대치하면

$$f*(Z) = \sum_{i=0}^{\infty} f(i\triangle)\,Z^{-i}$$

그림 4.3의 벽체의 실내측 벽면의 열류는 다음식으로 표시된다

$$q(L, s) = Y_{\delta}(s)\,T(0, s)$$

여기서 Y_{δ}가 전달함수이다. Z변환을 하면

$$q*(L, Z) = K(Z) \quad T*(0, Z)$$

여기서, K(Z)는 전달함수이며, Z^{-1} 의 다항식으로 표시된다.

$$K(Z) = \frac{N(Z)}{D(Z)} = \frac{b_o + b_1 Z^{-1} + b_2 Z^{-2} + \cdots}{d_o + d_1 Z^{-1} + d_2 Z^{-2} + \cdots}$$

따라서,

$$q(L,0)+q(L,\triangle)Z^{-1}+\cdot\cdot\cdot=\left[\frac{b_o+b_1Z^{-1}+\cdot\cdot\cdot}{d_o+d_1Z^{-1}+\cdot\cdot\cdot}\right]\cdot$$

$$[T(0,0)+T(0,\triangle)Z^{-1}+\cdot\cdot\cdot]$$

Z의 동차항의 계수를 같게 두면,

$$q(L,n\triangle)=\frac{1}{d_0}\left[\sum_{i=0}^{n}b_iT[0,(n-i)\triangle]-\sum_{i=1}^{n}d_iq[L,(n-i)\triangle]\right] \tag{4.21}$$

응답계수를 이용한 열전달은 다음식으로 표시되며, 서로 비교할 수 있다.

$$q(L,t)=\sum_{i=0}^{\infty}Y_iT(0,t-i\triangle) \tag{4.22}$$

Z변환에 의한 전달함수를 얻기 위해서는 램프 또는 계단형 여진의 라프라스 변환을 Z변환 한다. 그런 다음에 Z변환식의 출력인 O(Z)를 입력 I(Z)로 나누면 Z전달함수를 구할 수 있다.

전열행렬 $D(s)/B(s)=x\delta(s)$의 램프 입력을 라프라스 변환인 $\frac{1}{s^2}\cdot D(s)/B(s)$은 다음과 같이 표시된다.

$$X_r(s)=\frac{a}{s^2}+\frac{b}{s}+\sum_{i=1}^{\infty}\frac{c_i}{s+\beta_i} \tag{4.23}$$

여기서, $a=\frac{D(0)}{B(0)}$, $b=\frac{d}{ds}[\frac{D(s)}{B(s)}]_{s=0}$, $c_n=[\frac{D(s)}{s^2B'(s)}]_{s=-\beta_n}$

출력인 벽체의 전열응답인 식 (4.23)을 Z변환하면

$$O(Z)=X_r(Z)=\frac{a\triangle}{Z(1-Z^{-1})^2}+\frac{b}{1-Z^{-1}}+\sum_{i=1}^{\infty}\frac{c_i}{1-\exp(-\beta_i\triangle)Z^{-1}}$$

램프 입력인 $\frac{1}{s^2}\frac{D(s)}{B(s)}$의 Z변환 I(Z)는 다음과 같다.

$$I(Z) = \frac{\triangle}{Z(1-Z^{-1})^2}$$

전달함수 K(Z)를 N(Z)/D(Z)로 표시하면 출력은 다음과 같다.

$$O(Z) = K(Z)I(Z) = \frac{N(Z)}{D(Z)} I(Z)$$

전달함수의 분모의 계수 d_i는 다음식으로 표시된다.

$$D(Z) = \prod_{i=1}^{\infty} [1 - \exp(\beta_i \triangle)Z^{-1}] \tag{4.24}$$

분자 N(Z)에 상응하는 전달함수 b_i는 다음식으로 구한다.

$$N(Z) = \frac{Z(1-Z^{-1})^2}{\triangle} \prod_{i=1}^{\infty} (i - e^{-\beta_i \triangle} Z^{-1})(X_{r0} + X_{r1}Z^{-1} + X_{r2}Z^{-2} + \cdot\cdot\cdot) \tag{4.25}$$

식 (4.24)와 식 (4.25)를 정리하면 d_i, b_i는 표 4.3 및 4.4와 같다.

표 4.3 d_i 의 계산식

계수	계산식
d_0	1.0
d_1	$-[e^{-\beta_1}+e^{-\beta_2}+e^{-\beta_3}+\cdots\cdots+e^{-\beta_N}]$
d_2	$[e^{-(\beta_1+\beta_2)}+e^{-(\beta_1+\beta_3)}+e^{-(\beta_1+\beta_4)}+\cdots+e^{-(\beta_{N-1}+\beta_N)}]$
d_3	$-[e^{-(\beta_1+\beta_2+\beta_3)}+e^{-(\beta_1+\beta_2+\beta_4)}+\cdots+e^{-(\beta_{N-2}+\beta_{N-1}+\beta_N)}]$
d_4	$[e^{-(\beta_1+\beta_2+\beta_3+\beta_4)}+\cdots\cdots+e^{-(\beta_{N-3}+\beta_{N-2}+\beta_{N-1}+\beta_N)}]$
d_5	$-[e^{-(\beta_1+\beta_2+\beta_3+\beta_4+\beta_5)}+\cdots\cdots]$
d_6	$[e^{-(\beta_1+\beta_2+\beta_3+\beta_4+\beta_5+\beta_6)}+\cdots\cdots]$

표 4.4 b_i 의 계산식

계수	계산식
b_0	$A_0+\sum_{i=1}^{N}\frac{A_i}{\beta_i}(1-e^{-\beta_i})$
b_1	$A_0\,d_1+\sum_{i=1}^{N}\frac{A_i}{\beta_i}(1-e^{-\beta_i})\{d_1-(1-e^{-\beta_i})\}$
b_2	$A_0\,d_2+\sum_{i=1}^{N}\frac{A_i}{\beta_i}(1-e^{-\beta_i})\{d_2-d_1(1-e^{-\beta_i})-e^{-\beta_i}(1-e^{-\beta_i})\}$
b_3	$A_0\,d_3+\sum_{i=1}^{N}\frac{A_i}{\beta_i}(1-e^{-\beta_i})\{d_3-d_2(1-e^{-\beta_i})-d_1e^{-\beta_i}(1-e^{-\beta_i})-e^{-2\beta_i}(1-e^{-\beta_i})\}$
b_4	$A_0\,d_4+\sum_{i=1}^{N}\frac{A_i}{\beta_i}(1-e^{-\beta_i})\{d_4-d_3(1-e^{-\beta_i})-d_2e^{-\beta_i}(1-e^{-\beta_i})-d_1e^{-2\beta_i}(1-e^{-\beta_i})-e^{-3\beta_i}(1-e^{-\beta_i})\}$
b_5	$A_0\,d_5+\sum_{i=1}^{N}\frac{A_i}{\beta_i}(1-e^{-\beta_i})\{d_5-d_4(1-e^{-\beta_i})-d_3e^{-\beta_i}(1-e^{-\beta_i})-d_2e^{-2\beta_i}(1-e^{-\beta_i})-d_1e^{-3\beta_i}(1-e^{-\beta_i})-e^{-4\beta_i}(1-e^{-\beta_i})\}$
b_n	$A_0\,d_n+\sum_{i=1}^{N}\frac{A_i}{\beta_i}(1-e^{-\beta_i})\{d_n-d_{n-1}(1-e^{-\beta_i})-d_{n-2}e^{-\beta_i}(1-e^{-\beta_i})-d_{n-3}e^{-2\beta_i}(1-e^{-\beta_i})-d_{n-4}e^{-3\beta_i}(1-e^{-\beta_i})-e^{-(n-1)\beta_i}(1-e^{-\beta_i})\}$

(2) 알고리즘

'나'항의 응답계수와 동일하며 (다)항에서 전도전달함수를 표 4.3과 표 4.4로 계산 한다.

(3) 프로그램

부록 C에서 menu=3으로 두어 전도전달함수를 계산한다.

(가) 입력화면

입력화면은 그림 4.1과 같으며, 출력종류를 전도전달함수로 선정한다.

(나) 출력화면

1. 중간계산값

그림 4.4의 응답계수의 중간계산값과 같다.

2. 최종결과

그림 4.6은 전도전달함수의 최종결과 출력화면으로 b_n, d_n, $\sum C_n$ 및 열관류율 U의 값이다.

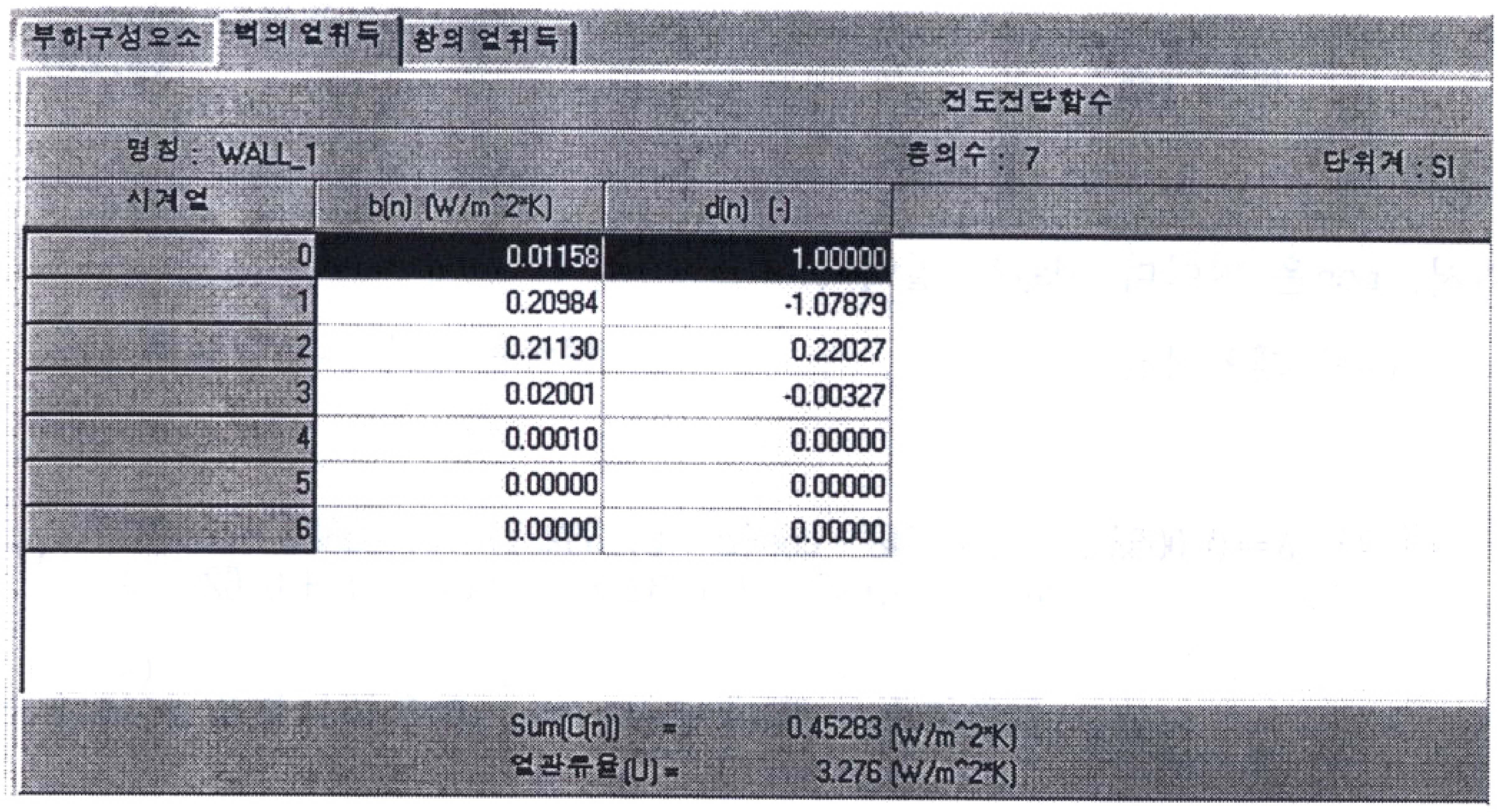


그림 4.6 전도전달함수의 출력화면

2. 그림자 계산 S/W개발

가. 설명

이 알고리즘은 벽과 창의 외부에 설치된 각종의 차양과 날개벽으로 인한 그림자의 길이를 구하여 유리창의 일사열취득에 필요한 일조면적율을 구하기 위한 것이다.

차양과 날개벽의 종류는 다양하나 격자루버형은 이들을 모두 포괄하고 있기 때문에 이것을 기준으로 알고리즘을 작성하였다.

그림자 길이를 계산하기 위해서는 그 도시의 위도와 경도, 계산시각, 벽의 방위각과 경사각 및 날개벽과 차양의 치수가 필요하다.

나. 알고리즘

(1) 일일정수를 계산한다.

(가) 년간 통산일수

$$n = 30 \cdot (mon - 1) + (mon + mon/8)/2 - (mon + 7)/10 + day \quad (4.26)$$

$$w = \frac{\pi \cdot n}{183} \quad (4.27)$$

여기서, mon은 월이며, day는 일이다.

(나) 태양적위

$$\delta = 0.006322 - 0.405748 \cdot \cos(w + 0.153231) - 0.005880 \cdot \cos(2.0 \cdot w + 0.207099) - 0.003233 \cdot \cos(3.0 \cdot w + 0.620129) \quad \text{(rad)} \quad (4.28)$$

(다) 균시차

$$\varepsilon = -0.000279 + 0.122772 \cdot \cos(w + 1.49831) - 0.165458 \cdot \cos(2.0 \cdot w - 1.26155) - 0.005354 \cdot \cos(3.0w - 1.571) \quad \text{(h)} \quad (4.29)$$

(라) 시차 = 진태양시 - 지방표준시

$$\Delta t = \varepsilon + (LONG - 135)/15 \ (h) \qquad (4.30)$$

여기서, LONG는 그 지방의 경도이다.

(2) 시각(時角)을 계산한다.

$$HA = (t - 12 + \Delta t) \cdot \pi/12$$

(3) 태양위치를 계산한다.

(가) 태양고도

$$\sin\beta = \cos L \cdot \cos\delta \cdot \cos HA + \sin L \cdot \sin\delta \qquad (4.31)$$

여기서, L은 그지방의 위도이다.

(나) 태양방위각

$$\cos As = (\sin\beta \cdot \sin L - \sin\delta)/(\cos\beta \cdot \cos L) \qquad (4.32)$$

(4) 벽태양방위각을 계산한다.

$$\gamma = As - Aw \qquad (4.33)$$

여기서 Aw는 벽방위각이며, γ는 다음과 같이 수정한다.

$\gamma = \gamma - 360$, $\gamma \geq 180$ 일때

$\gamma = \gamma + 360$, $\gamma \leq -180$ 일때

(5) 태양입사각을 계산한다.

$$\cos\theta = \sin\beta \cdot \cos\alpha + \cos\beta \cdot \sin\alpha \cdot \cos\gamma \qquad (4.34)$$

여기서, α는 벽의 경사각이다.

만약 cos θ ≤ 0 이면 유리창일조율 $SR_{win}=0$, 벽일조율 $SR_{wal}=0$으로 처리한다

(6) 형상각을 계산한다.

(가) 수직면

$$\tan\gamma_v = \tan\gamma \qquad (4.35)$$

$$\tan\phi_v = \tan\beta / \cos r$$

여기서, $\tan\phi_v$는 형상각이며, $\tan\gamma_v$는 벽태양방위각의 정현이다.

(나) 경사면

$$\tan\gamma = \tan\gamma_v / (\sin\alpha + \tan\phi_v \cdot \cos\alpha) \qquad (4.36)$$

$$\tan\phi = (-\cos\alpha + \tan\phi_v \cdot \sin\alpha) / (\sin\alpha + \tan\phi_v \cdot \cos\alpha)$$

(7) 날개벽의 그림자 길이, XL을 계산한다.

(가) $Tan\ \gamma \le 0$ 일때

$$a_1 = x_4$$

$$a_3 = x_2$$

$$XL = -\tan\gamma \cdot x_1$$

(나) $Tan\ \gamma > 0$ 일 때

$$a_1 = x_2$$

$$a_3 = x_4$$

$$XL = \tan\gamma \cdot x_5$$

(8) 차양에 의한 그림자길이 YL를 계산한다.

(가) $\tan\phi$ ≤ 0 일때

$b_1 = y_4$

$b_3 = y_2$

YL = - tanϕ · y_5

(나) tanϕ > 0 일때

$b_1 = y_2$

$b_3 = y_4$

YL = - tanϕ · y_1

(9) 전일조면적 W1을 계산한다.

(가) 벽의 가로 X_w > XL , 또는 벽의 세로길이 Y_w > YL이면

$SR_{win} = SR_{wal} = 0$

(나) 전일조면적 W1계산

$W1 = (X_w - XL) \cdot (Y_w - YL)$

(10) 유리창 일조면적 W3 및 벽일조면적 W4 계산

(가) 유리창만 그늘진 경우, XL ≥ ($x_3 + a_3$)

유리창 일조면적 W3 = 0

벽 일조면적 W4 = W1

(나) 유리창의 일조면적 가로

ⓐ 그림자가 창에 미달된 경우, XL ≤ a_3

W2 = x_3

ⓑ 창이 일부 그늘진 경우

W2 = x_3 + a_3 - XL

(다) 차양에 의해 유리창이 그늘진 경우, YL ≥ ($b_1 + y_3$)

W3 = 0

W4 = W1

(라) 차양 그림자가 유리창에 미달된 경우, YL ≤ b_1

W3 = y_3 · W2

W4 = W1 - W3

(마) 차양 그림자가 창에 도달한 경우, YL > b_1

W3 = ($b_1 + y_3$-YL) · W2

W4 = W1 - W3

(11) 유리창 일조면적율 SR_{win} 및 벽일조면적율 SR_{wal} 계산

SR_{win} = W3/($x_3 \cdot y_3$)

SR_{wal} = W4 / A_{wal}

다. 프로그램

(1) 프로그램의 주요 계산부는 부록 D와 같다.

(가) 입력화면

그림 4.7은 입력화면의 일예이며, 그림자를 계산하기위한 4가지 입력데이터의 입력방법은 다음과 같다.

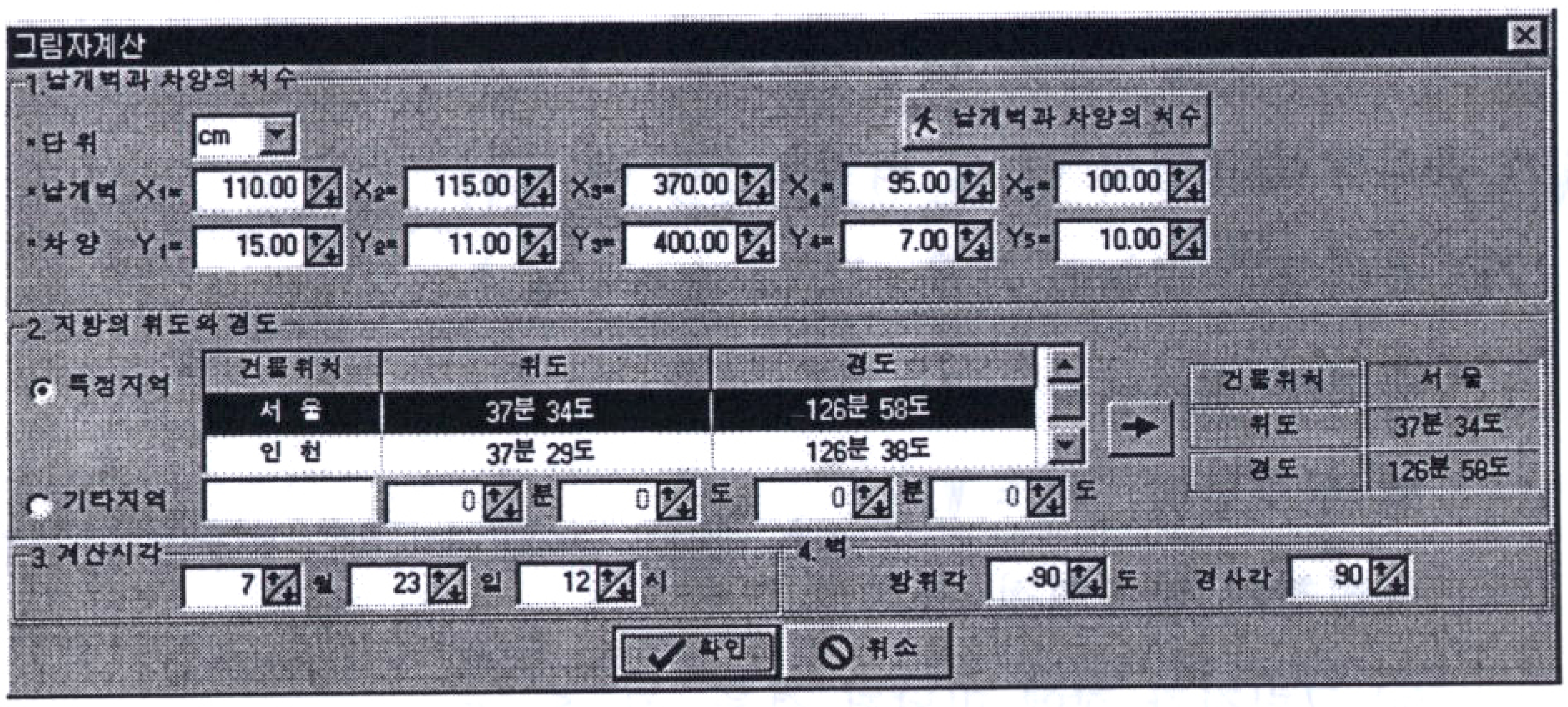


그림 4.7 입력화면의 예

1. 날개벽과 차양의 치수

그림 4.8은 외부차폐물 및 벽과 태양의 위치를 위한 도움화면이며, 차양과 날개벽은 다음과 같이 입력한다.

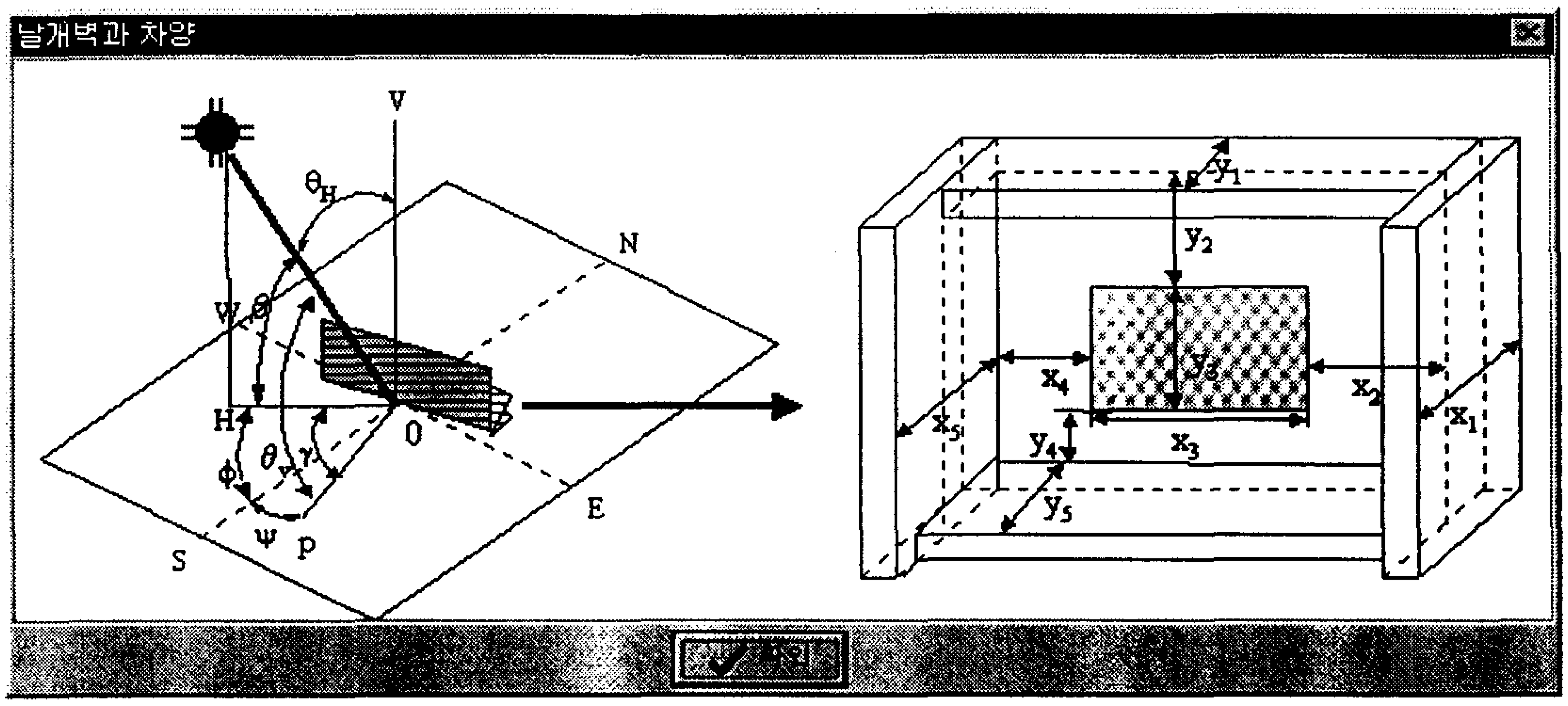


그림 4.8 벽과 태양의 위치 및 차양과 날개벽의 치수

가. 모든 벽면이 창인 경우

- $X_2 = X_4 = 0$, $Y_2 = Y_4 = 0$

나. 창이 없고 모두 벽인 경우

- $X_3 = 0$, $Y_3 = 0$
- $X_2 + X_4$ = 벽의 가로, $Y_2 + Y_4$ = 벽의 세로가 되도록 치수를 부여한다.
- 치수의 단위는 m, cm, mm 중 어떤것이라도 좋지만 X_i, Y_i 의 단위는 같아야 한다.

2. 도시의 위도와 경도

도시의 위도와 경도입력은 사용자의 편의를 위해 표 4.5와 같이 국내 26개 대도시의 위도와 경도값을 스크롤박스안에서 선택하여 값을 설정할 수 있도록 하였고, 기타도시를 선택할 경우에는 사용자가 직접 건물위치, 위도 및 경도를 입력할 수 있도록 되어있다.

표 4.5 주요도시의 위도와 경도 및 표고

도시	위도 Lat. (N)	경도 Long. (E)	표고 Height (m)
속 초	38° 12′	128° 36′	25.8
대관령	37° 41′	128° 44′	820.0
춘 천	37° 54′	127° 44′	74.0
강 릉	37° 45′	128° 54′	26.0
서 울	37° 34′	126° 58′	85.5
인 천	37° 29′	126° 38′	68.9
울릉도	37° 29′	130° 54′	221.1
수 원	37° 16′	126° 59′	36.9
서 산	36° 47′	126° 27′	19.7
울 진	36° 59′	129° 25′	49.5
청 주	36° 38′	127° 26′	59.0
대 전	36° 18′	127° 24′	77.1
추풍령	36° 13′	128° 00′	245.9
포 항	36° 02′	129° 23′	5.6
군 산	36° 59′	126° 42′	26.3
대 구	35° 53′	128° 37′	57.8
전 주	35° 49′	127° 09′	51.2
울 산	35° 33′	129° 19′	31.5
광 주	35° 08′	126° 55′	70.9
부 산	35° 06′	129° 02′	69.2
충 무	34° 50′	128° 26′	32.2
목 포	34° 47′	126° 23′	53.4
여 수	34° 44′	127° 44′	67.0
제 주	33° 31′	126° 32′	22.0
서귀포	33° 14′	126° 34′	51.9
진 주	35° 12′	128° 06′	21.5

3. 계산시각

계산시각의 입력시 각각의 범위는 다음과 같다.

- 월 : 1월 ~ 12월
- 일 : 1일 ~ 각 달의 마지막날
- 시간 : 1시 ~ 24시

4. 벽의 방위각 및 경사각

벽방위각은 정남(S)으로부터 시계방향으로 측정한다.

반시계방향으로 측정할 경우에는 (-)부호를 붙인다.

(나) 출력화면

그림 4.9는 출력화면으로 화면의 좌측에 그림자의 형상 그리고 우측에 주요 계산

값들로 구성되어 있다.

각 입력값을 설정후 계산한 결과, 프로그램상의 출력결과는 그림 4.9와 같으며 태양고도(β)는 70.779°, 태양방위각(A_s)은 -29.549°, 벽태양방위각(γ)은 60.451° 이고, 그림자의 가로방향길이(X)는 176.398 ㎝, 세로방향의 길이(Y)는 87.238 ㎝로 계산되었다.

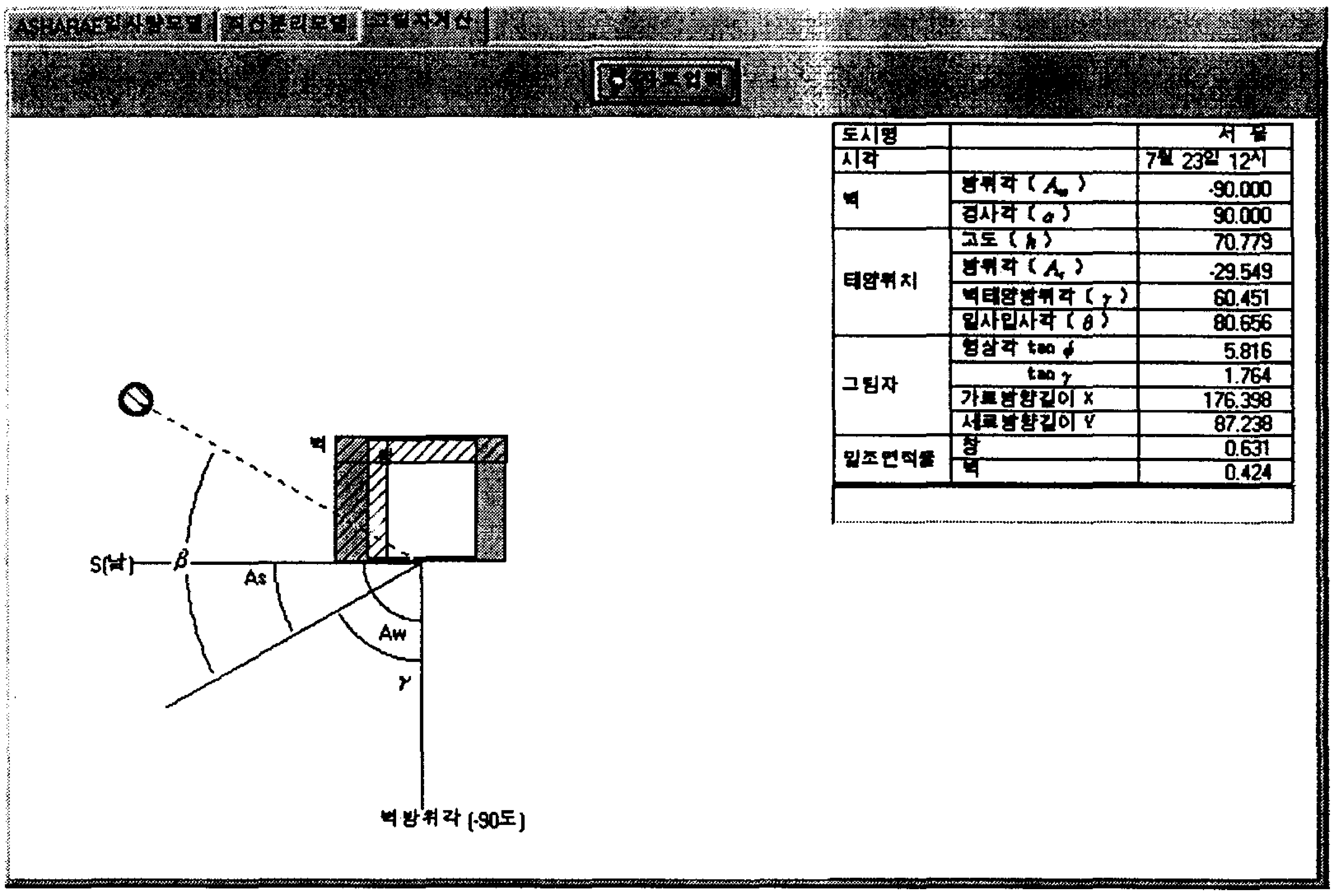


그림 4.9 계산예제의 입력화면

3. 기기 특성의 회귀분석

가. 설명

공조시스템의 시뮬레이션에는 열원시스템을 구성하고 있는 각종기기의 특성치를 수식화할 필요가 있다. 구성기기의 특성치는 보통 제조회사의 카타로그에 제시되어 있으나 데이터가 이산적으로 주어지는 경우가 많기 때문에 이산적인 값사이에 있는 임의의 점에서의 값을 추정하기 위해서는 이산적인 데이터를 함수형으로 나타낼 필요가 있다. 여기서는 데이터에 관련된 오차량에 기초한 최소제곱회귀분석(least-squares regression)을 이용하여 독립변수가 하나인 다항식의 회귀분석

(polynomial regression)과 독립변수가 2개인 이중선형(bi-linear)식과 이중이차(bi-quadratic)식에 대한 회귀분석 프로그램을 개발하였다.

나. 알고리즘

(1) 다항식의 회귀분석(polynomial regression)

Y가 X의 함수로 표현되는 데이터의 직선 혹은 곡선적합은 최소제곱회귀분석을 이용한다. 데이터가 (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ······· ,(X_n, Y_n) 일 때 m차 다항식으로 적합시키도록 다음과 같이 확장시킬 수 있다.

$$Y = C_1 + C_2X + C_3X^2 + \cdots\cdots\cdots + C_{m+1}X^m + e \qquad (4.37)$$

여기서 e 는 오차를 나타낸다. 오차의 제곱합은 다음과 같다.

$$S = \sum_{i=1}^{n} e_i^{\,2} = \sum_{i=1}^{n} (Y_i - C_1 - C_2X_i - C_3X_i - \cdots\cdots\cdots - C_{m+1}X^m)^2 \qquad (4.38)$$

계수 C1, C2, C3, ·····,C_{m+1}를 결정하기 위하여 식(4.38)를 각각의 계수에 대하여 편미분하면 다음과 같다.

$$\frac{\partial S}{\partial C_1} = -2\sum_{i=1}^{n}(Y_i - C_1 - C_2X_i - C_3X_i^{\,2} - \cdots\cdots - C_{m+1}X_i^{\,m})$$

$$\frac{\partial S}{\partial C_2} = -2\sum_{i=1}^{n}X_i(Y_i - C_1 - C_2X_i - C_3X_i^{\,2} - \cdots\cdots - C_{m+1}X_i^{\,m})$$

$$\frac{\partial S}{\partial C_3} = -2\sum_{i=1}^{n}X_i^{\,2}(Y_i - C_1 - C_2X_i - C_3X_i^{\,2} - \cdots\cdots - C_{m+1}X_i^{\,m}) \qquad (4.39)$$

$$\vdots$$

$$\frac{\partial S}{\partial C_{m+1}} = -2\sum_{i=1}^{n}X_i^{\,m}(Y_i - C_1 - C_2X_i - C_3X_i^{\,2} - \cdots\cdots - C_{m+1}X_i^{\,m})$$

오차의 제곱의 합, S가 최소가 되려면 식(4.39)이 각각 0이어야 하므로 정리하면 다음과 같다.

$$
\begin{aligned}
nC_1 &+ (\sum_{i=1}^{n} X_i)C_2 + (\sum_{i=1}^{n} X_i^{2})C_3 + \cdots\cdots + (\sum_{i=1}^{n} X_i^{m})C_{m+1} &= \sum_{i=1}^{n} Y_i \\
(\sum_{i=1}^{n} X_i)C_1 &+ (\sum_{i=1}^{n} X_i^{2})C_2 + (\sum_{i=1}^{n} X_i^{3})C_3 + \cdots\cdots + (\sum_{i=1}^{n} X_i^{m+1})C_{m+1} &= \sum_{i=1}^{n} X_i Y_i \\
(\sum_{i=1}^{n} X_i^{2})C_1 &+ (\sum_{i=1}^{n} X_i^{3})C_2 + (\sum_{i=1}^{n} X_i^{4})C_3 + \cdots\cdots + (\sum_{i=1}^{n} X_i^{m+2})C_{m+1} &= \sum_{i=1}^{n} X_i^{2} Y_i \qquad (4.40) \\
\cdot & \qquad \cdot \qquad \cdot \qquad \cdot & \cdot \\
\cdot & \qquad \cdot \qquad \cdot \qquad \cdot & \cdot \\
\cdot & \qquad \cdot \qquad \cdot \qquad \cdot & \cdot \\
(\sum_{i=1}^{n} X_i^{m})C_1 &+ (\sum_{i=1}^{n} X_i^{m+1})C_2 + (\sum_{i=1}^{n} X_i^{m+2})C_3 + \cdots\cdots + (\sum_{i=1}^{n} X_i^{2m})C_{m+1} &= \sum_{i=1}^{n} X_i^{m} Y_i
\end{aligned}
$$

또는 행렬식으로 다음과 같다.

$$
\begin{bmatrix}
n & \Sigma X & \Sigma X^2 & \cdots\cdots & \Sigma X^m \\
\Sigma X & \Sigma X^2 & \Sigma X^3 & \cdots\cdots & \Sigma X^{m+1} \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\Sigma X^m & \Sigma X+1 & \Sigma X^{m+2} & \cdots\cdots & \Sigma X^{2m}
\end{bmatrix}
\begin{bmatrix}
C_1 \\ C_2 \\ \cdot \\ \cdot \\ C_{m+1}
\end{bmatrix}
=
\begin{bmatrix}
\Sigma Y \\ \Sigma XY \\ \cdot \\ \cdot \\ \Sigma X^m Y
\end{bmatrix}
\qquad (4.41)
$$

여기서 Σ 은 $\sum_{i=1}^{n}$ 이고, X, Y는 X_i Y_i 이며 이하 같다.

이 행렬식으로부터 계수는 다음과 같이 구해진다.

$$
\begin{bmatrix}
C_1 \\ C_2 \\ \cdot \\ \cdot \\ C_{m+1}
\end{bmatrix}
=
\begin{bmatrix}
n & \Sigma X & \Sigma X^2 & \cdots\cdots & \Sigma X^m \\
\Sigma X & \Sigma X^2 & \Sigma X^3 & \cdots\cdots & \Sigma X^{m+1} \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\Sigma X^m & \Sigma X^{m+1} & \Sigma X^{m+2} & \cdots\cdots & \Sigma X^{2m}
\end{bmatrix}^{-1}
\begin{bmatrix}
\Sigma Y \\ \Sigma XY \\ \cdot \\ \cdot \\ \Sigma X^m Y
\end{bmatrix}
\qquad (4.42)
$$

추정치의 표준오차는 다음과 같이 정의된다.

$$S_{Y/X} = \sqrt{\frac{S}{n-(m+1)}} \qquad (4.43)$$

최적적합(best fit)은 m에 따른 다항식별로 오차의 제곱합이 가장 적은 차식을 선택한다.

(2) 이중선형식과 이중이차식

Z가 X, Y인 두개의 독립변수의 함수중 이중선형식인 경우에 다음과 같이 확장시킬 수 있다.

$$Z = C_1 + C_2X + C_3Y + e \qquad (4.44)$$

여기서 C1, C2, C3의 최적 값을 구하기 위하여 오차의 제곱합은 다음과 같다.

$$S = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (Z_i - C_1 - C_2X_i - C_3Y_i)^2 \qquad (4.45)$$

계수 C1, C2, C3를 결정하기 위하여 식 (4.45)를 각각의 계수에 대하여 편미분하면 다음과 같다.

$$\frac{\partial S}{\partial C_1} = -2\sum_{i=1}^{n}(Z_i - C_1 - C_2X_i - C_3Y_i)$$

$$\frac{\partial S}{\partial C_2} = -2\sum_{i=1}^{n}[(Z_i - C_1 - C_2X_i - C_3Y_i)(-X_i)] \qquad (4.46)$$

$$\frac{\partial S}{\partial C_3} = -2\sum_{i=1}^{n}[(Z_i - C_1 - C_2X_i - C_3Y_i)(-Y_i)]$$

오차의 제곱의 합, S 가 최소가 되려면 식(4.46)이 각각 0 이어야 하므로 정리하면 다음과 같다.

$$nC_1 \quad + (\sum_{i=1}^{n} X_i)C_2 + (\sum_{i=1}^{n} Y_i)C_3 = \sum_{i=1}^{n} Z_i$$

$$(\sum_{i=1}^{n} X_i)C_1 + (\sum_{i=1}^{n} X_i^2)C_2 + (\sum_{i=1}^{n} X_iY_i)C_3 = \sum_{i=1}^{n} X_iZ_i \tag{4.47}$$

$$(\sum_{i=1}^{n} Y_i)C_1 + (\sum_{i=1}^{n} X_iY_i)C_2 + (\sum_{i=1}^{n} Y_i^2)C_3 = \sum_{i=1}^{n} Y_iZ_i$$

또는 행렬식으로 다음과 같다.

$$\begin{bmatrix} n & \Sigma X & \Sigma Y \\ \Sigma X & \Sigma X^2 & \Sigma XY \\ \Sigma Y & \Sigma XY & \Sigma Y^2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} \Sigma Z \\ \Sigma XZ \\ \Sigma YZ \end{bmatrix} \tag{4.48}$$

식 (4.48)에서 식 (4.42)와 같은 방법으로 각 계수를 구할 수 있다.

Z가 X, Y인 두개의 독립변수의 함수중 이중이차식인 경우에 다음과 같이 확장시킬 수 있다.

$$Z = C_1 + C_2X + C_3X^2 + C_4Y + C_5Y^2 + C_6XY + e \tag{4.49}$$

오차의 제곱합을 각 계수에 대하여 편미분하여 정리하면 다음과 같다.

$$\begin{bmatrix} n & \Sigma X & \Sigma X^2 & \Sigma Y & \Sigma Y^2 & \Sigma XY \\ \Sigma X & \Sigma X^2 & \Sigma X^3 & \Sigma XY & \Sigma XY^2 & \Sigma X^2Y \\ \Sigma X^2 & \Sigma X^3 & \Sigma X^4 & \Sigma X^2Y & \Sigma X^2Y^2 & \Sigma X^3Y \\ \Sigma Y & \Sigma XY & \Sigma X^2Y & \Sigma Y^2 & \Sigma Y^3 & \Sigma XY^2 \\ \Sigma Y^2 & \Sigma XY^2 & \Sigma X^2Y^2 & \Sigma Y^3 & \Sigma Y^4 & \Sigma XY^3 \\ \Sigma XY & \Sigma X^2Y & \Sigma X^3Y & \Sigma XY^2 & \Sigma XY^3 & \Sigma X^2Y^2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{bmatrix} = \begin{bmatrix} \Sigma Z \\ \Sigma XZ \\ \Sigma X^2Z \\ \Sigma YZ \\ \Sigma Y^2Z \\ \Sigma XYZ \end{bmatrix} \tag{4.50}$$

식 (4.50)에서 식 (4.42)와 같은 방법으로 각 계수를 구할 수 있다.

Z가 X, Y인 두개의 독립변수의 이중이차식의 다른 경우로 다음과 같이 확장시킬 수 있다.

$$Z = C_1 + C_2X + C_3X^2 + C_4Y + C_5Y^2 + C_6XY + C_7X^2Y + C_8XY^2 + C_9X^2Y^2 + e \qquad (4.51)$$

오차의 제곱합을 각 계수에 대하여 편미분하여 정리하면 다음과 같다.

$$\begin{bmatrix} n & \Sigma X & \Sigma X^2 & \Sigma Y & \Sigma Y^2 & \Sigma XY & \Sigma X^2Y & \Sigma XY^2 & \Sigma X^2Y^2 \\ \Sigma X & \Sigma X^2 & \Sigma X^3 & \Sigma XY & \Sigma XY^2 & \Sigma X^2Y & \Sigma X^3Y & \Sigma X^2Y^2 & \Sigma X^3Y^2 \\ \Sigma X^2 & \Sigma X^3 & \Sigma X^4 & \Sigma X^2Y & \Sigma X^2Y^2 & \Sigma X^3Y & \Sigma X^4Y & \Sigma X^3Y^2 & \Sigma X^4Y^2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \quad \begin{bmatrix} \Sigma Z \\ \Sigma XZ \\ \Sigma X^2Z \end{bmatrix}$$

$$\begin{bmatrix} \Sigma Y & \Sigma XY & \Sigma X^2Y & \Sigma Y^2 & \Sigma Y^3 & \Sigma XY^2 & \Sigma X^2Y^2 & \Sigma XY^4 & \Sigma X^2Y^3 \\ \Sigma Y^2 & \Sigma XY^2 & \Sigma X^2Y^2 & \Sigma Y^3 & \Sigma Y^4 & \Sigma XY^3 & \Sigma X^2Y^3 & \Sigma XY^4 & \Sigma X^2Y^4 \\ \Sigma XY & \Sigma X^2Y & \Sigma X^3Y & \Sigma XY^2 & \Sigma XY^3 & \Sigma X^2Y^2 & \Sigma X^3Y^2 & \Sigma X^2Y^3 & \Sigma X^3Y^3 \end{bmatrix} \begin{bmatrix} C_4 \\ C_5 \\ C_6 \end{bmatrix} = \begin{bmatrix} \Sigma YZ \\ \Sigma Y^2Z \\ \Sigma XYZ \end{bmatrix} \qquad (4.52)$$

$$\begin{bmatrix} \Sigma X^2Y & \Sigma X^3Y & \Sigma X^4Y & \Sigma X^2Y^2 & \Sigma X^2Y^3 & \Sigma X^3Y^2 & \Sigma X^4Y^2 & \Sigma X^3Y^3 & \Sigma X^4Y^3 \\ \Sigma XY^2 & \Sigma X^2Y^2 & \Sigma X^3Y^2 & \Sigma XY^3 & \Sigma XY^4 & \Sigma X^2Y^3 & \Sigma X^3Y^3 & \Sigma X^2Y^4 & \Sigma X^3Y^4 \\ \Sigma X^2Y^2 & \Sigma X^3Y^2 & \Sigma X^4Y^2 & \Sigma X^2Y^3 & \Sigma X^2Y^4 & \Sigma X^3Y^3 & \Sigma X^4Y^3 & \Sigma X^3Y^4 & \Sigma X^4Y^4 \end{bmatrix} \begin{bmatrix} C_7 \\ C_8 \\ C_9 \end{bmatrix} \quad \begin{bmatrix} \Sigma X^2YZ \\ \Sigma XY^2Z \\ \Sigma X^2Y^2 \end{bmatrix}$$

식 (4.52)에서 식 (4.42)와 같은 방법으로 각 계수를 구할 수 있다.

다. 프로그램

본 연구에서 1차적으로 선정 또는 개발한 공조기기의 특성치 회귀분석 프로그램은 표 4.6과 같으며 앞으로 지수형 등의 최적적합 회귀분석프로그램을 추가로 개발하여 통합하고 윈도우 GUI 환경으로 개발할 것이다.

프로그램 ASHCURVE.FOR는 참고문헌 9)에 있으며, BICURVE.FOR는 부록 E와 같다.

표 4.5 회귀분석 프로그램

방정식의 형식	독립변수	방정식	프로그램명
다항식	1개	$y = c_1 + c_2x + c_3x^2 + \cdot\cdot\cdot + c_nx^{n-1}$	ASHCURVE.FOR
이중선형식	2개	$z = c_1 + c_2x + c_3y$	BICURVE.FOR
이중이차식	2개	$z = c_1 + c_2x + c_3x^2 + c_4y + c_5y^2 + c_6xy$	BICURVE.FOR
이중4차식	2개	$z = c_1 + c_2x + c_3x^2 + c_4y + c_5y^2 + c_6xy + c_7x^2y + c_8xy^2 + c_9x^2y^2$	ASHCURVE.FOR

주 : ASHCURVE.FOR는 참고문헌 9), pp. 19 - 29 참조

그림 4.10은 이중이차식에 대한 입력데이터이며, 그림 4.11은 출력 데이터이다.

```
Size of matrix (NxN)
6,3,1,7,BIQUAD1
1.    1.    21.
1.    2.    46.
1.6   4.   146.
2.    8.   465.
5.    5.   380.
6.    6.   540.
8.    9.  1080.
END
```

그림 4.10 이중이차식의 입력 데이터

```
NxN matrix =    6
no. of columns of data =    3
no. of columns for solution vector =    1
no. of data points =    7
type of equation = BIQUAD1

 equation form

      Z = C(1) + C(2)*X +  C(3)*X**2
          + C(4)*Y + C(5)*Y**2 + C(6)*X*Y

Coefficients are
******************

C( 1)  =     1.459602
C( 2)  =     1.315707
C( 3)  =     3.267852
C( 4)  =     4.167701
C( 5)  =     5.042867
C( 6)  =     5.734606

      x           y          input  z     calculated  z

 .1000E+01   .1000E+01   .2100E+02   .2099E+02
 .1000E+01   .2000E+01   .4600E+02   .4602E+02
 .1600E+01   .4000E+01   .1460E+03   .1460E+03
 .2000E+01   .8000E+01   .4650E+03   .4650E+03
 .5000E+01   .5000E+01   .3800E+03   .3800E+03
 .6000E+01   .6000E+01   .5400E+03   .5400E+03
 .8000E+01   .9000E+01   .1080E+04   .1080E+04

STANDARD ERROR IS =   .9766E-03
```

그림 4.11 이중이차식의 출력 데이터

제 5 절 공조방식 알고리즘 작성

1. 공조방식의 개요

가. 공조방식의 분류

(1) 단일구역 공조방식(Single Zone A/C System)

한개의 방 또는 여러개의 방이라도 부하변동 특성이 비슷해서 통합하여 취급할 수 있는 공조공간을 대상으로 하는 공조기 및 공조방식을 단일 구역 공조시스템이라 한다. 공조공간의 냉난방 부하는 시간에 따라 그 크기와 현열・잠열구성 비율이 달라 지는데 이에 맞추어 냉각, 가열용량을 조절해 주어야 한다. 단일구역 공조방식에서는 각각의 방이 동일한 부하변동 특성을 가지므로 단순히 그 시각에 필요로 하는 냉각 또는 가열량을 공조기에서 처리해 주기만 하면 된다. 즉, 냉방과 난방이 동시에 필요한 경우는 없으며 각각의 방이 거의 일정한 비율로 냉난방 부하가 증가 또는 감소한다고 가정하는 것이다. 따라서 혼합, 재열에 따른 에너지 손실도 전혀 없으며 제어도 아주 단순하다.

(2) 복합구역 공조방식(Multi-Zone A/C System)

거의 대부분의 건물에 있어서는 사용조건(인원밀도, 조명부하, 사용시간대)등이 다르고 방위와 개구율(창문면적)이 틀린 여러개의 방이 있으며 이들의 부하변동 특성도 다양하다. 그러므로 단일구역 공조방식으로는 이러한 여러 개의 방들을 통합하여 쾌적한 온습도 조건으로 유지하는 것이 불가능하며 또 각각의 방에 대해 하나하나 단일구역 공조방식을 적용하여 수 많은 공조기를 설치하는 것도 비현실적이다. 이러한 이유로 여러가지 복잡한 공조방식이 개발되어 있는데 예를 들면 단말재열방식(Terminal Reheat), 이중덕트 방식(Double Duct), 변풍량 방식(Variable Air Volume), 물-공기 병용 방식(Air-Water System) 등등이 있다. 이렇게 부하특성이 다른 여러 공조공간을 동시에 처리하기 위한 복잡한 공조방식을 복합구역 공조방식(Multi-Zone A/C System)이라고 한다.

나. 공조방식의 선택

위에서 말한 여러가지 공조방식은 각각의 특징과 장단점, 사용상의 한계를 가지고 있으므로 공조방식을 최적 선정하기 위해서는 다음의 여러 사항들을 충분히 고려 하여야 한다.

* 최적 시스템의 선택을 위한 고려사항
  - 냉, 난방 열원의 종류
  - 기계실, 천정속 덕트 공간의 크기
  - 실내공간의 활용성
  - 온습도 제어 성능
  - IAQ (실내 공기 청정도)
  - 초기 설치비
  - 운전비 (Energy Cost)
  - 유지관리의 용이성
  - 소음, 누수 위험

본 연구에서는 그림 5.1에 나타낸 바와 같이 국내외에서 보편적으로 사용하고 있는 16개의 공조방식에 대한 계산 절차를 분석, 정리 하였다.

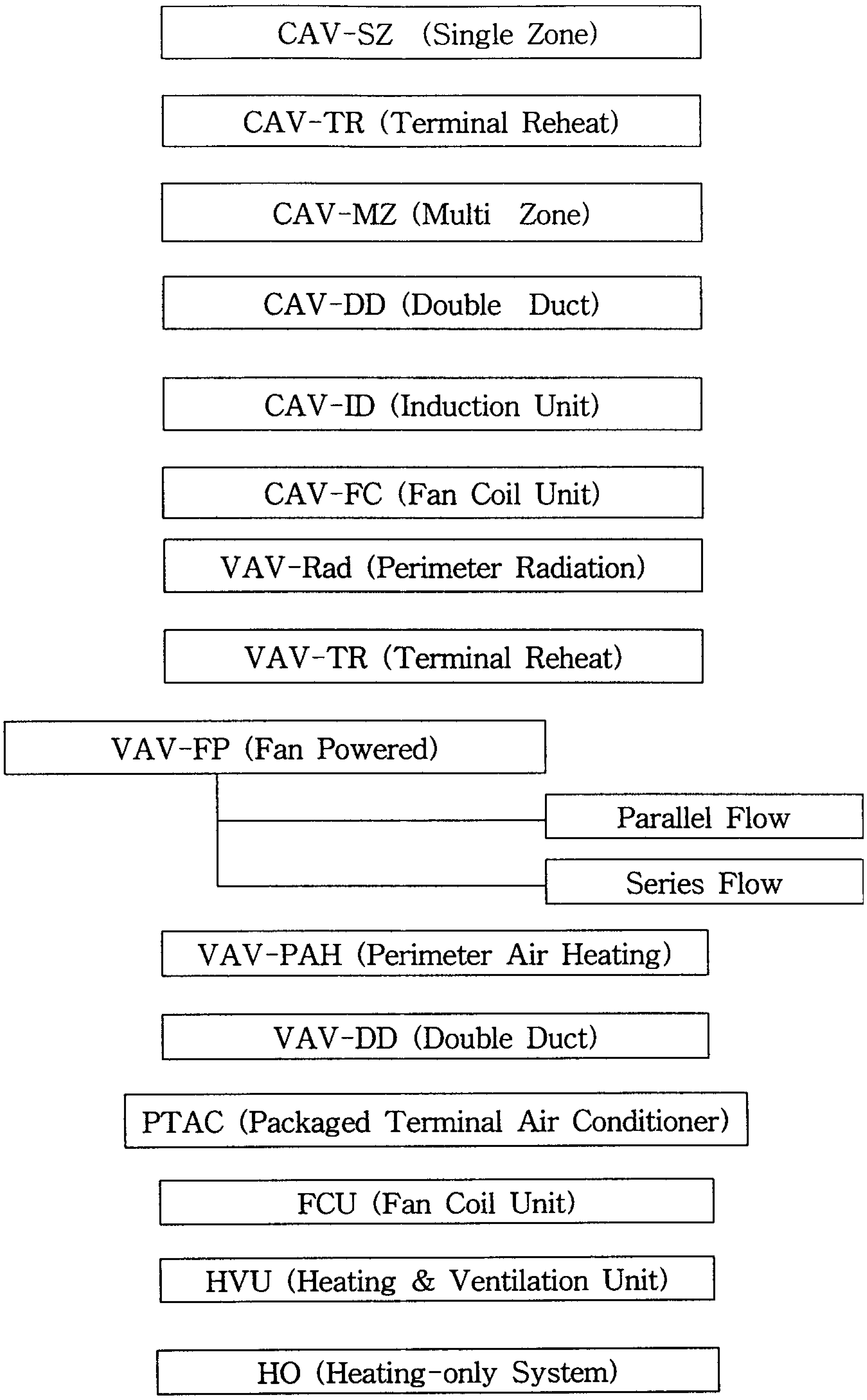


그림 5.1 공조방식의 종류

2. Room 및 Zone의 계산으로부터 참조되는 함수

가. ROOM의 부하계산 결과로부터 참조되는 함수

함 수 명	설 명	단위
NR	실의 입력 번호 (Rec. Number)	(정수)
NZ	소속터미널 번호	(정수)
NS	소속 공조기 번호	(정수)
RMAR(NR)	실의 면적 (가로 x 세로)	(㎡)
RMVOL(NR)	실의 체적 (가로 x 세로 x 높이)	(㎥)
RMPP(NR)	실내인원수	(인)
RMMT(NR)	동일 space의 갯수	(정수)
RMACH(NR)= "a/c/h"	실내공조범위 a = 냉난방(all) c = 냉방(cooling) h = 난방(heating)	(-)
RMSAMI(NR)	실의 최소급기풍량 = (RMSAVMIXNG(NR) × RMVOL(NR)) / 3.6 또는 = RMSAVMIAR(NR) × RMAR(NR)	(ℓ/s)
RMEAV(NR)	실내배기량 = RMEAVLPS (배기량지정) 또는 = (RMEAVXNG(NR) × RMVOL(NR)) / 3.6	(ℓ/s)
RMSAEA(NR)="y/n"	y = 급기풍량은 실내배기량 이상으로 한다 n = 급기풍량은 실내배기량과 무관	(-)
RMQS(NR, N)	각 N(시각)에 있어서의 실내냉방현열부하	(W)
RMQL(NR, N)	각 N(시각)에 있어서의 실내냉방잠열부하	(W)
RMQT(NR, N)	각 N(시각)에 있어서의 실내총부하 = RMQS(N) + RMQL(N)	(W)
RMQPLM(NR, N)	각 N(시각)에 있어서의 실내냉방 플레넘 부하	(W)
RMQH(NR)	실내난방부하	(W)
RMQHUM(NR)	실내 가습부하 (kg/s)	(kg/s)
RMQPN(NR)	RMQS(NR, N)가 최대값으로 되는 시각	(시)
RMQPS(NR)	Room Peak N에 있어서의 실내냉방현열부하	(W)
RMQPL(NR)	Room Peak N에 있어서의 실내냉방잠열부하	(W)
RMQPT(NR)	= RMQPS(NR) + RMQPL(NR)	(W)
RMQPPLM(NR)	Room Peak N에 있어서의 실내냉방 플레넘 부하	(W)
RMQPSHF(NR)	Room Peak N에 있어서의 실내현열비 = RMQPS(NR) / RMQPT(NR) (실수 0~1사이)	(-)

나. Zone(터미날유니트)에 관련된 함수

함 수 명	설 명	단위
NZ	존의 번호	(정수)
ZOMT(NZ)	동일 Zone의 개수 (정수)	(정수)
ZOPP(NZ)	존의 인원수 = Σ (RMPP(NR) × RMMT(NR))	(인)
ZOQS(NZ, N)	존의 현열부하 합계 $= \sum_{NR} (RMQS(NR, N) \times RMMT(NR))$	(W)
ZOQL(NZ, N)	존의 잠열부하 합계 = Σ (RMQL(NR, N) × RMMT(NR))	(W)
ZOQT(NZ, N)	존의 총부하 = ZOQS(NZ, N) + ZOQL(NZ, N)	(W)
ZOQPL(NZ, N)	존의 플레넘부하 합계 = Σ (RMQPLM(NR, N) × RMMT(NR))	(W)
ZOQH(NZ)	존의 난방부하 = Σ (RMQH(NR) × RMMT(NR))	(W)
ZOQHUM(NZ)	존의 가습부하 = Σ (RMQHUM(NR) × RMMT(NR))	(W)
ZOAR(NZ)	존의 면적 = Σ (RMAR(NR) × RMMT(NR))	(m^2)
ZOVOL(NZ)	존의 체적 = Σ (RMVOL(NR) × RMMT(NR))	(m^3)
ZOSAMI(NZ)	Zone(터미널)의 최소급기풍량(ℓ/ s) = (ZOSAMIXNG(NZ) × ZOVOL(NZ)) / 3.6 또는 = ZOSAMIAR(NZ) × ZOAR(NZ)	(ℓ/s)
ZOEAV(NZ)	존의 배기량 = Σ (RMEAV(NR) x RMMT(NR))	(ℓ/s)
ZOQPN(NZ)	ZOQS(NZ, N)가 최대값으로 되는 N(시각)	(시)
ZOQPS(NZ)	Zone Peak N에 있어서의 Zone의 냉방현열부하 = [ZOQS(NZ, N)]max	(W)
ZOQPL(NZ)	Zone Peak N에 있어서의 Zone의 냉방잠열부하 = ZOQL(NZ, ZOQPN)	(W)
ZOQPT(NZ)	존의 총부하 = ZOQPS(NZ) + ZOQPL(NZ)	(W)
ZOQPPLM(NZ)	Zone Peak 시간에 있어서의 Zone의 냉방Plnm부하 = [ZOQPLM(NZ, N)]max	(W)
ZOQPSHF(NZ)	= ZOQPS(NZ) / ZOQPT(NZ) (Zone Peak N에 있어서의 실내현열비)	(-)

다. Air System에 관련된 함수

함 수 명	설 명	단위
NS	시스템의 번호	(정수)
SYMT(NS)	동일 공조기의 수	(개)
SYAR(NS)	시스템의 면적 = Σ (RMAR(NR) × RMMT(NR) × ZOMT(NZ))	(㎡)
SYVOL(NS)	시스템의 체적 = Σ (RMVOL(NR) × RMMT(NR) × ZOMT(NZ))	(㎥)
SYTPP(NS)	시스템의 인원수 = Σ (RMPP(NR) × RMMT(NR) × ZOMT(NZ))	(인)
SYQS(NS, N)	시스템의 현열부하 $= \sum_{NR}$ (RMQS(NR, N) × RMMT(NR) × ZOMT(NZ))	(W)
SYQL(NS, N)	시스템의 잠열부하 = Σ (RMQL(NR, N) × RMMT(NR) × ZOMT(NZ))	(W)
SYQT(NS, N)	시스템의 총부하 = SYQS(NS, N) + SYQL(NS, N)	(W)
SYPLQ(NS, N)	시스템의 플레넘부하 = Σ (RMQPLM(NR, N) × RMMT(NR) × ZOMT(NZ))	(W)
SYQH(NS)	시스템의 난방부하 = Σ (RMQH(NR) × RMMT(NR) × ZOMT(NZ))	(W)
SYQHUM(NS)	시스템의 가습부하 = Σ (RMQHUM(NR) × RMMT(NR) × ZOMT(NZ))	(kg/s)
SYEA(NS)	시스템의 배기량 = Σ (RMEAV(NR) × RMMT(NR) × ZOMT(NZ))	(ℓ/s)
SYQPSN(NS)	SYQS(NS, N)가 최대값으로 되는 N(시각)	(시)
SYQPS(NS)	System Peak N에 있어서의 System의 냉방현열부하 = [SYQS(NS, N)]max	(W)
SYQPL(NS)	System Peak N에 있어서의 System의 냉방잠열부하 = SYQL(NS, SYQPN)	(W)
SYQPT(NS)	시스템의 최대 냉방부하 = SYQPS(NS) + SYQPL(NS)	(W)
SYQPPLM(NS)	System Peak N에 있어서의 System의 냉방Plnm부하	(W)
SYQPSHF(NS)	= SYQPS(NS) / SYQPT(NS) (System Peak N에 있어서의 실내현열비)	(-)

라. 정풍량(CAV) 방식에서의 공조기 관련 함수

(1) Zone 계산에 필요한 Block Load

함 수 명	설 명	단위
ZOBKQS(NZ)	존의 최대현열부하 합계 = Σ (RMQPS(NR) × RMMT(NR))	(W)
ZOBKQL(NZ)	존의 최대잠열부하 합계 = Σ (RMQPL(NR) × RMMT(NR))	(W)
ZOBKQPLM(NZ)	존의 최대플레넘부하 합계 = Σ (RMQPPLM(NR) × RMMT(NR))	(W)
ZOBKQT(NZ)	존의 최대총부하 = ZOBKQS(NZ) + ZOBKQL(NZ)	(W)
ZOBKSHF(NZ)	존의 현열비 = ZOBKQS(NZ) / ZOBKQT(NZ)	(-)

(2) Air System 계산에 필요한 Block Load

함 수 명	설 명	단위
SYBKQS(NS)	시스템의 현열부하 = Σ(RMQPS(NR)×RMMT(NR)×ZOMT(NZ))	(W)
SYBKQL(NS)	시스템의 잠열부하 = Σ(RMQPL(NR)×RMMT(NR)×ZOMT(NZ))	(W)
SYBKQPLM(NS)	시스템의 플레넘부하 = Σ(RMQPPLM(NR)×RMMT(NR)×ZOMT(NZ))	(W)
SYBKQT(NS)	시스템의 총부하 = SYBKQS(NS) + SYBKQL(NS)	(W)
SYBKSHF(NS)	시스템의 현열비 = SYBKQS(NS) / SYBKQT(NS)	(-)

마. 변풍량(VAV) 방식에서의 공조기 관련 함수

(1) Zone 계산에 필요한 Block Load

변 수 명	설 명	단위
ZOBKQS(NZ)	존의 현열부하 =[Σ(RMQS(NR, N) × RMMT(NR))] max * Room & N에 대해 최대값을 찾는다 * 최대값일 때의 N를 ZOBKN로 한다.	(W)
ZOBKQL(NZ)	존의 잠열부하 =Σ(RMQL(NR, ZOBKN) × RMMT(NR))	(W)
ZOBKPLM(NZ)	존의 플레넘부하 = Σ(RMQPLM(NR, ZOBKN) × RMMT(NR))	(W)
ZOBKT(NZ)	존의 총부하 = ZOBKQS(NZ) + ZOBKQL(NZ)	(W)
ZOBKSHF(NZ)	존의 현열비 = ZOBKQS(NZ) / ZOBKQT(NZ)	(-)

(2) Air System 계산에 필요한 Block Load

변 수 명	설 명	단위
SYBKQS(NS)	Peak시각에 대한 시스템 현열부하 = [Σ (RMQS(NR, N) × RMMT(NR) × ZOMT(NZ))] max * Room & N에 대해 최대값을 찾는다 * 최대값일 때의 N를 SYBKN로 한다.	(W)
SYBKQL(NS)	Peak시각에 대한 시스템 잠열부하 = Σ(RMQL(NR, SYBKN) × RMMY(NR) ×ZOMT(NZ))	(W)
SYBKPLM(NS)	Peak시각에 대한 시스템 플레넘부하 = Σ(RMQPLM(NR, SYBKN) × RMMT(NR) × ZOMT(NZ))	(W)
SYBKQT(NS)	시스템의 총부하 = SYBKQS(NS) + SYBKQL(NS)	(W)
SYBKSHF(NS)	시스템의 현열비 = SYBKQS(NS) / SYBKQT(NS)	(-)

3. 공조방식의 알고리즘에 사용된 변수

변 수	내 용	단 위
ACF	공기밀도 수정계수	-
ACSAV	공조기 급기풍량	ℓ/s
CCLADB	냉각코일 출구온도	℃
CCLARH	냉각코일출구공기의 상대습도	%
CCLARHMX	냉각코일출구공기의 최대상대습도	%
CCLAEP	냉각코일출구공기의 엔탈피	kJ/kg
CCLAWB	냉각코일출구공기의 습구온도	℃
CCLAX	냉각코일출구공기의 절대습도	g/kg
CLADT	냉방취출온도차	℃
EAFV	배기팬 풍량	ℓ/s
EAV	배기 풍량	ℓ/s
HLADT	난방취출온도차	℃
HUMTR	가습에 의한 온도상승	℃
HUMTW	물/전기/통풍기화식 수온	℃
MADBH2	수정혼합공기 건구온도	℃
MAEPH2	수정혼합공기 엔탈피	kJ/kg
MARHH2	수정혼합공기 상대습도	%
MAWBH2	수정혼합공기 습구온도	℃
MAXH2	수정혼합공기 절대습도	g/kg
NR	실의 입력 번호 (Rec. Number)	-
NS	소속 공조기 번호	-
NZ	존의 번호	-
OADBC2	수정외기건구온도	℃
OADBH	외기 건구온도	℃
OADBH2	수정외기건구온도	℃
OAEPH2	수정전열교환기통과후의 엔탈피	kJ/kg
OARHH2	수정전열교환기통과후의 상대습도	%
OAVAR	면적당 외기도입량	(ℓ/s)/㎡
OAV	외기 도입량	ℓ/s
OAVPP	일인당 외기도입량	ℓ/s
OAVSA	급기풍량의 비율에의한 외기도입량	ℓ/s

(계속)

(계속)

변 수	내 용	단 위
OAVSA	급기풍량의 비율에의한 외기도입량	ℓ/s
OAXNG	환기횟수에 의한 외기도입량	ℓ/s
OAWBH2	수정전열교환기통과후의 습구온도	℃
OAXC2	수정외기 절대습도	g/kg
OAX	외기 절대습도	g/kg
OAXH2	수정전열교환기통과후의 절대습도	g/kg
P	표고 Z(m)에 따라 계산된 대기압	kPa
PHCLADB	예열코일출구온도	℃
PHCSTKGS	예열코일 증기용량	kg/s
PHCELKW	예열코일 전기용량	W
PO	표준대기압	kPa
QLHEEF	잠열교환효율	-
QPH	예열열량	W
QSAF	급기 FAN에의한 부하	W
QSHEEF	현열교환효율	-
RADB	환기건구온도	℃
RADBC(RN)	실별부하에서입력한 실내온도	℃
RADTR	RA(환기)덕트 온도상승	℃
RAFEF	FAN 효율	-
RAMOEF	MOTOR 효율	-
RAFSP	FAN 정압	kPa
RAFTR	팬에 의한 급기온도 상승	℃
RAFV	환기팬 풍량	ℓ/s
RAPLTR	플레넘부하에 의한 RA의 온도상승	℃
RAV	환기풍량	ℓ/s
RAX	환기절대습도	g/kg
RMCLADT(RN)	취출온도차	℃
RMDX	실내부하에 의한 절대습도 상승	℃
RMEAV(RN)	실내배기량	ℓ/s
RMQPS	실내최대현열부하	ℓ/s
RMRAV	실내에서 환기되는 풍량	ℓ/s
RMSAV(RN)	실내 급기풍량	ℓ/s
RMSAVMI(RN)	실내 최소급기량	ℓ/s
SADBC	냉풍덕트온도	℃
SADB	급기 건구온도	℃

(계속)

(계속)

변 수	내 용	단 위
SADTR	급기 덕트의 온도상승	℃
SAFEF	FAN 효율	-
SAFMOEF	MOTOR 효율	-
SAFSP	급기 FAN 정압	kPa
SAFTR	급기 팬에 의한 온도상승	℃
SAVC	냉방 급기 풍량	ℓ/s
SAVH	난방 급기풍량	ℓ/s
STP	증기분사압력	kPa
STQL	증기잠열	W
SYAR	시스템의 면적	㎡
SYBKQL	시스템 잠열	W
SYBKQL	시스템 현열	W
SYEA	시스템의 배기량	ℓ/s
SYHUMGS	가습량	kg/s
SYPLQ	시스템의 플레넘 부하	W
SYQH	시스템난방부하	W
SYRMDBC	공조기에서 입력한 여름철 실내대표온도조건	℃
SYRMDBH	공조기에서 입력한 겨울철 실내대표온도조건	℃
SYRMXC	시스템 대표실내 냉방 절대습도	g/kg
SYTPP	시스템의 총인원수	인
SYV	시스템의 총풍량	ℓ/s

4. 공조방식의 계산에 필요한 함수

가. Draw-Thru Fan에 의한 온도 상승 계산(F_SAFTR)

(1) 계산 알고리즘

(가) 급기(SA) Fan의 위치 선택 (SAFanType = 0, 1)

- DRAW - THRU (선택)
- BLOW - THRU

(나) 급기(SA) Fan의 온도상승 고려사항 선택

1. 온도상승고려안함 일때

SAFTR = 0

2. 온도상승고려 일때

- 입력사항

변수명	명칭	단위
SAFSP	FAN 정압	kPa
SAFEF	FAN 효율	%

3. 모터의 위치 선택

가. 공기의 흐름내(Inside AirStream)에 위치할 때

- 입력사항

변수명	명칭	단위
SAFMOEF	MOTOR효율	%

$$SAFTR = 0.81 \times SAFSP \times \frac{100}{SAFEF} \times \frac{100}{SAFMOEF}$$

나. 공기의 흐름밖(Out of AirSteam)에 위치할 때

$$SAFTR = 0.81 \times SAFSP \times \frac{100}{SAFEF}$$

(2) 반환값

(가) 팬에 의한 급기온도 상승 : SAFTR

(나) 급기 FAN에 의한 부하

QSAF = 1.23 x ACF × ACSAVC × SAFTR

(3) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
SAFSP	FAN 정압	kPa	2	0.1	1.2	
SAFEF	FAN 효율	%	100	10	75	
SAFMOEF	MOTOR 효율	%	100	10	80	

(나) 출력변수

변수명	명칭	단위	비 고
SAFTR	팬에 의한 급기온도 상승	℃	
QSAF	급기 FAN에의한 부하	W	

(다) 내부변수

변수명	명칭	단위	비 고
ACF	공기밀도 수정계수	-	
ACSAVC	공조기 급기풍량	ℓ/s	

나. Blow-Thru Fan에 의한 온도상승 계산(F_SAFTR)

(1) 계산 알고리즘

(가) 급기(SA) Fan의 위치 선택

- DRAW - THRU
- BLOW - THRU (선택)

(나) 급기(SA) Fan의 온도상승 고려사항 선택

1. 온도상승고려안함 일때

SAFTR = 0

2. 온도상승고려 일때

- 입력사항

변수명	명칭	단위
SAFSP	FAN 정압	kPa
SAFEF	FAN 효율	%

3. 모터의 위치 선택

가. 공기의 흐름내(Inside AirStream)에 위치할 때

- 입력사항

변수명	명칭	단위
SAFMOEF	MOTOR효율	%

$$SAFTR = 0.81 \times SAFSP \times \frac{100}{SAFEF} \times \frac{100}{SAFMOEF}$$

나. 공기의 흐름밖(Out of AirSteam)에 위치할 때

$$SAFTR = 0.81 \times SAFSP \times \frac{100}{SAFEF}$$

(2) 반환값

(가) 팬에 의한 급기온도 상승 : SAFTR

(나) 급기 FAN에 의한 부하

QSAFan = 1.23 x ACF × ACSAVC × SAFTR

(3) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
SAFSP	FAN 정압	kPa	2	0.1	1.2	
SAFEF	FAN 효율	%	100	10	75	
SAFMOEF	MOTOR 효율	%	100	10	80	

(나) 출력변수

변수명	명칭	단위	비 고
SAFTR	팬에 의한 급기온도 상승	℃	
QSAF	급기 팬에 의한 부하	W	

(다) 내부변수

변수명	명칭	단위	비 고
ACF	공기밀도 수정계수	-	
ACSAVC	공조기 급기풍량	ℓ/s	

다. 환기(RA) Fan에 의한 온도상승(F_RAFTR)

(1) 계산 알고리즘

(가) Fan의 상태 선택

- 없음 -> RAFTR = 0
- 환기팬
- 배기팬 -> RAFTR = 0

(나) 환기팬 선택시

1. 온도상승고려안함 일때

RAFTR = 0

2. 온도상승고려 일때

- 입력사항

명칭	변수명	단위
FAN 정압	RAFSP	kPa
FAN 효율	RAFEF	%

3. 모터의 위치 선택

가. Inside AirStream 일때

- 입력사항

명칭	변수명	단위
MOTOR 효율	RAFMOEF	%

$$RAFTR = 0.81 \times RAFSP \times \frac{100}{RAFEF} \times \frac{100}{RAFMOEF}$$

나. Out of AirSteam 일때

$$RAFTR = 0.81 \times RAFSP \times \frac{100}{RAFEF}$$

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비고
RAFSP	FAN 정압	kPa	2	0.1	1.2	
RAFEF	FAN 효율	%	100	10	75	
RAFMOEF	MOTOR 효율	%	100	10	80	

(나) 출력변수

변수명	명칭	단위	비고
RAFTR	팬에 의한 급기온도 상승	℃	
QRAF	급기 FAN에 의한 부하	W	

(다) 내부변수

변수명	명칭	단위	비고
ACF	공기밀도 수정계수	-	
ACSAVC	공조기 급기풍량	ℓ/s	

라. 급기 및 환기의 덕트에 의한 온도상승 계산(F_SADTR, F_RADTR)

(1) 계산 알고리즘

(가) 입력 사항

변 수 명	명 칭	단 위
SADTR	SA(급기)덕트 온도상승	℃
RADTR	RA(환기)덕트 온도상승	℃

(나) 출력값

1. 급기 덕트에 의한 부하

QSAD = 1.23 x ACF × ACSAVC × SADTR (W)

2. 환기 덕트에 의한 부하

QRAD = 1.23 x ACF × ACRAVC × RADTR (W)

마. 냉방풍량 결정(냉방취출온도차, CLADT 기준)(F_SAVC)

(1) 계산 알고리즘

(가) 입력사항

변 수 명	명 칭	단 위
CLADT	냉방취출온도차	℃
CCLARHMX	냉각코일출구 공기의 상대습도	%

(나) 냉방풍량 계산

$$SAVC = \frac{SYBKQS}{1.23 \times ACF \times CLADT}$$

(다) 냉방급기온도 계산

SADBC = SYRMDBC - CLADT

(라) 냉각코일출구온도 계산

CCLADB = SADBC - (SAFTR + SADTR)

(마) 실내부하에 의한 절대습도 상승 계산

$$RMDX = \frac{SYBKQL}{3.01 \times ACF \times SAVC}$$

(바) 냉각코일 출구공기의 상태 계산

SAXC = SYRMXC - RMDX

CCLAX = SAXC

CCLADB, CCLAX → CCLARH, CCLAWB, CCLAEP

(습공기계산)

(사) 만약 CCLARH > CCLARHMX 이면

CLADT = CLADT - 0.1 and GOTO (1) (반복계산)

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비고
CLADT	냉방취출온도차	℃	2	20	10	
CCLARHMX	냉각코일출구 공기의 상대습도	%	60	95	90	

(나) 출력변수

변수명	명칭	단위	비고
SAVC	냉방풍량	ℓ/s	
CLADT	냉방취출온도차	℃	
SADBC	냉방급기온도	℃	
CCLADB	냉각코일 출구온도	℃	
RMDX	실내부하에 의한 절대습도 상승	g/kg	
CCLARH, CCLAWB, CCLAEP, CCLADB, CCLAX	냉각코일 출구공기의 상태	-	

(다) 내부변수

변수명	명칭	단위	비고
ACF	공기밀도 수정계수	-	
SYBKQS	시스템 현열	W	
SYBKQL	시스템 잠열	W	
SYRMDBC	시스템 대표실내 냉방 건구온도	℃	
SAFTR	급기 팬의 온도상승	℃	
SADTR	급기 덕트의 온도상승	℃	
SAVC	냉방 급기 풍량	ℓ/s	
SYRMXC	시스템 대표실내 냉방 절대습도	g/kg	

바. 냉방풍량 결정(냉풍덕트온도, SADBC 기준)(F_SAVC)

(1) 계산 알고리즘

(가) 입력사항

변 수 명	명 칭	단 위
SADBC	냉풍 덕트 온도	℃
CCLARHMX	냉각코일출구 공기의 상대습도	%

(나) 냉방취출 온도차의 계산

CLADT = RMDBC - SADBC

(다) 냉방풍량 계산

$$SAVC = \frac{SYBKQS}{1.23 \times ACF \times CLADT}$$

(라) 냉방급기온도 계산

SADBC = SYRMDBC - CLADT

(마) 냉각코일출구온도 계산

CCLADB = SADBC - (SAFTR + SADTR)

(바) 실내부하에 의한 절대습도 상승(g/kg)

$$RMDX = \frac{SYBKQL}{3.01 \times ACF \times SAVC}$$

(사) 냉각코일 출구공기의 상태 계산

SAXC = SYRMXC - RMDX

CCLAX = SAXC

CCLADB, CCLAX → CCLARH, CCLAWB, CCLAEP

(아) 만약 CCLARH > CCLARHMX 이면

CLADT = CLADT - 0.1 and GOTO (1) (반복계산)

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비고
SADBC	냉풍덕트온도	℃	25	5	20	
CCLARHMX	냉각코일출구 공기의 상대습도	%	95	60	90	

(나) 출력변수

변수명	명칭	단위	비고
SAVC	냉방풍량	ℓ/s	
CLADT	냉방취출온도차	℃	
SADBC	냉방급기온도	℃	
CCLADB	냉각코일 출구온도	℃	
RMDX	실내부하에 의한 절대습도 상승	g/kg	
CCLARH, CCLAWB, CCLAEP, CCLADB, CCLAX	냉각코일 출구공기의 상태	-	

(다) 내부변수

변수명	명칭	단위	비고
ACF	공기밀도 수정계수	-	
SYBKQS	시스템 현열	W	
SYBKQL	시스템 잠열	W	
SYRMDBC	시스템 대표실내 냉방 건구온도	℃	
SAFTR	급기 팬의 온도상승	℃	
SADTR	급기 덕트의 온도상승	℃	
SAVC	냉방 급기 풍량	ℓ/s	
SYRMXC	시스템 대표실내 냉방 절대습도	g/kg	

사. 각 ROOM별 풍량 수정계산(F_RMRSAV)

(1) 계산 알고리즘

(가) 입력사항

명 칭	변 수 명	단 위
냉방취출 온도차	CLADT	℃
공조기에서 입력한 실내대표온도	SYRMDBC	℃
실별부하에서 입력한 실내온도	RMDBC(NR)	℃

(나) 실내 급기풍량 계산

$$RMCLADT(NR) = CLADT - SYRMDBC + RMDBC(NR)$$

$$RMSAVC(NR) = \frac{RMQPS(NR)}{1.23 \times ACF \times RMCLADT(NR)}$$

(다) 만약 RMSAVC(NR) < RMSAVMI(NR) 이면

RMRSAVC(NR) = RMSAVMI(NR)

(라) 만약 RMSAEA(NR) = "y" 이고 RMSAVC(NR) < RMEAV(NR) 이면

RMSAVC(NR) = RMEAV(NR)

(2) 변수명

(가) 입력변수

명칭	변수명	단위	최대	최소	기본값	비 고
냉방취출 온도차	CLADT	(℃)	20	2	10	
공조기에서 입력한 실내대표온도	SYRMDBC	(℃)	30	20	26	
실별부하에서 입력한 실내온도	RMDBC(NR)	(℃)	30	20	26	

(나) 출력변수

변수명	명칭	단위	비 고
RMSAVC(NR)	실내 급기풍량	ℓ/s	

(다) 내부변수

변수명	명칭	단위	비 고
RMCLADT(NR)	취출온도차	℃	
RMQPS(NR)	실내최대현열부하	W	
RMSAVMI(NR)	실내최소급기량	ℓ/s	
RMSAEA(NR)	y or n	ℓ/s	
RMEAV(NR)	실내배기량	ℓ/s	

아. 여름 외기량 계산(F_OAVC)

(1) 계산 알고리즘

(가) 입력사항

1. 외기도입량 입력 및 선택

- 일인당 외기도입량에 의한 방법 : 여름 OAVPPC (ℓ/s)/인
- 면적당 도입량에 의한 방법 : 여름 OAVARC (ℓ/s)/㎡
- 환기횟수에 의한 방법 : 여름 OAXNGC x/Hr
- 급기풍량의 비율에 의한 방법 : 여름 OAVSAC %

2. 외기도입량 결정기준 선택

· 최대, · 최소

3. 실내 배기량 기준 선택

· YES, · NO

(나) 여름외기량 산정기준 입력으로 부터 각각의 외기량 계산기준에 따라 계산하고, 계산값의 최대, 최소의 선택에 따라 외기량을 계산한다.

OAVPPC	(ℓ/s)/인 × SYTPP	max or min
OAVARC	(ℓ/s)/m² × SYAR	
OAXNGC	x/Hr × SYV / 3.6	
OAVSAC	% × ACSAVC / 100	

(다) 실내배기량 기준이 YES 이고 OAVVC < SYEAV 이면,

OAVC = SYEAV

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비고
OAVPPC	일인당 외기도입량	ℓ/s	10	0	7	
OAVARC	면적당 외기도입량	(ℓ/s)/m²	10	0	5	
OAXNGC	환기횟수에 의한 방법	x/Hr	10	0	7	
OAVSAC	급기풍량의 비율에 의한 방법	%	100	0	30	

(나) 출력변수

변수명	명칭	단위	비고
OAVC	외기도입량	ℓ/s	

(다) 내부변수

변수명	명칭	단위	비고
SYTPP	시스템의 총인원수	인	
SYAR	시스템의 면적	m²	
SYV	시스템의 풍량	ℓ/s	
ACSAVC	공조기 풍량	ℓ/s	
SYEAV	시스템의 배기량	ℓ/s	

자. 겨울 외기량 계산(F_OAVH)

(1) 계산 알고리즘

(가) 입력사항

1. 외기도입량 입력 및 선택

- 일인당 외기도입량에 의한 방법 : 겨울 OAVPPH (ℓ/s)/인
- 면적당 도입량에 의한 방법 : 겨울 OAVARH (ℓ/s)/㎡
- 환기횟수에 의한 방법 : 겨울 OAXNGH x/Hr
- 급기풍량의 비율에 의한 방법 : 겨울 OAVSAH %

2. 외기도입량 결정기준 선택

- 최대, · 최소

3. 실내 배기량 기준 선택

- YES, · NO

(나) 겨울외기량 산정기준 입력으로 부터 각각의 외기량 계산기준에 따라 계산하고, 계산값의 최대, 최소의 선택에 따라 외기량을 계산한다.

OAVPPH	(ℓ/s)/인 × SYTPP	max or min
OAVARH	(ℓ/s)/㎡ × SYAR	
OAXNGH	x/Hr × SYV / 3.6	
OAVSAH	% × ACSAVC / 100	

(다) 실내배기량 기준이 YES 이고 OAVH < SYEAV 이면,

OAVC = SYEAV

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비고
OAVPPH	일인당 외기도입량	ℓ/s	10	0	7	
OAVARH	면적당 외기도입량	(ℓ/s)/㎡	10	0	6	
OAXNGH	환기횟수에 의한 방법	x/Hr	10	0	8	
OAVSAH	급기풍량의 비율에 의한 방법	%	100	0	20	

(나) 출력변수

변수명	명칭	단위	비고
OAVC	외기도입량	ℓ/s	

(다) 내부변수

변수명	명칭	단위	비고
SYTPP	시스템의 총인원수	인	
SYAR	시스템의 면적	㎡	
SYV	시스템의 풍량	ℓ/s	
ACSAVC	공조기 풍량	ℓ/s	
SYEAV	시스템의 배기량	ℓ/s	

차. 환기량 계산(F_RAVC)

(1) 계산 알고리즘

(가) 공조기풍량(ACSAVC)과 외기량(OAVC)을 읽어온다

(나) 환기량 계산

RAVC = ACSAVC - OAVC

(2) 변수명

(가) 출력변수

변수명	명 칭	단 위	비 고
RAVC	환기량	ℓ/s	

(나) 내부변수

변수명	명 칭	단 위	비 고
ACSAVC	공조기 풍량	ℓ/s	
OAVC	외기 도입량	ℓ/s	

카. 환기 및 배기 Fan의 풍량 계산(F_RAFVC, F_EAFVC)

(1) 계산 알고리즘

(가) 팬의 형태 선택

· NONE, · 환기팬, · 배기팬

(나) (가)에서 NONE을 선택한 경우

RAFVC = 0 , EAVC = 0 , EAFVC = 0

(다) (가)에서 환기팬을 선택한 경우

RAFVC = ACSAVC -SYEAV

EAFVC = 0 ※ (RAV + EAV = RAFV 이므로)

EAVC = RAFVC - RAVC

여기서, 실내공기압은 평형을 이루고 있다고 가정함

즉, 실내가 양압(Positive)이나 부압(Negative)으로 되지 않고

±0의 균형(=대기압)을 이루고 있다고 가정함

(라) (가)에서 배기팬을 선택한 경우

RAFVC = 0

(RAV + EAFV = ACSAV - SYEAV 이므로

EAV = EAFV = ACSAV - SYEAV - RAV

여기서, RAV = SAV - OAV 이므로)

EAFVC = OAVC - SYEAV, EAVC = EAFVC

(2) 변수명

(가) 출력변수

변수명	명칭	단위	비 고
RAFVC	환기팬 풍량	ℓ/s	
EAVC	배기 풍량	ℓ/s	
EAFVC	배기팬 풍량	ℓ/s	

(나) 내부변수

명칭	변수명	단위	비 고
공조기 급기풍량	ACSAVC	ℓ/s	
시스템 배기량	SYEAV	ℓ/s	
환기풍량	RAVC	ℓ/s	

파. ROOM → 환기(RA)로 나가는 풍량 계산(F_RMRAVC)

(1) 계산 알고리즘

(가) 풍량계산

RMRAVC = RAVC + EAVC

(나) 환기 및 배기 Fan의 선택에 따른 계산

- NONE : RMRAVC = RAVC
- 환기팬 : RMRAVC = RAFVC
- 배기팬 : RMRAVC = ACSAVC - SYEAV

  RAF, RAD, RAPLM 등에서의 온도상승 계산시 기준풍량은 RMRAVC 이다.

(2) 변수명

(가) 출력변수

변수명	명칭	단위	비 고
RMRAVC	실에서 RA로 나가는 풍량	ℓ/s	

(나) 내부변수

변수명	명칭	단위	비 고
RAVC	환기 풍량	ℓ/s	
RAFVC	환기팬의 풍량	ℓ/s	
ACSAVC	공조기 급기 풍량	ℓ/s	
SYEAV	시스템 배기량	ℓ/s	

하. 플레넘부하에 의한 환기(RA)의 온도상승(F_RAPLMTR)

(1) 계산 알고리즘

$$RAPLMTR(N) = \frac{SYPLQ(N)}{1.23 \times ACF \times RMRAVC}$$

(2) 변수명

(가) 출력변수

변수명	명칭	단위	비 고
RAPLTR(N)	플레넘부하에 의한 RA의 온도상승	℃	

(나) 내부변수

변수명	명 칭	단 위	비 고
SYPLMQ(N)	시스템의 플레넘 부하	W	
RMRAVC	실내에서 환기되는 풍량	ℓ/s	
ACF	공기밀도 수정계수	-	

거. 전열교환기(여름)의 외기처리 계산(F_HEOAC)

(1) 계산 알고리즘

(가) 기상자료에서 OADBC(N), OAXC(N)를 읽어 와서

OARHC(N), OAWBC(N), OAEPC(N) 습공기 계산

(나) 전열교환기의 성능 DATA 입력

현열교환효율 : QSHEEFC (%)

잠열교환효율 : QLHEEFC (%)

(다) RA또는 EA의 조건을 구한다.

(배기 Fan에 의한 온도상승은 무시한다.

즉, 배기 Duct에 의한 온도상승은 RADuct 온도상승에 포함되어 있는 것으로 본다.)

RADBC(N)=SYRMDBC + RAPLTR(N) + RADTR + RAFTR

RAXC = SYRMXC

RADBC(N), RAXC 로부터 습공기 상태량 계산

→ RARHC(N), RAWBC(N), RAEPC(N)

EADBC(N) = RADBC(N)

EAXC = RAXC

(라) 전열교환기의 외기처리 계산

OADBC2(N) = OADBC(N) - QSHEEFC × (OADBC(N) - EADBC(N)) / 100

OAXC2 = OAXC - QLHEEFC × (OAXC - EAXC) / 100

OADBC2(N), OAXC2 로부터 습공기 상태량 계산

→ OARHC2(N), OAWBC2(N), OAEPC2(N)

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
QSHEEFC	현열교환효율	%	100	0	55	
QLHEEFC	잠열교환효율	%	100	0	65	

(나) 출력변수

1. 외기의 상태량

OADBC(N), OAXC(N), OARHC(N), OAWBC(N), OAEPC(N)

2. 환기의 상태량

RADBC(N), RAXC(N), RARHC(N), RAWBC(N), RAEPC(N)

3. 수정외기의 상태량

OADBC2(N),OAXC(N),OAWBC2(N), OAEP2(N)

너. 전열교환기(겨울)의 외기처리 계산(F_HEOAH)

(1) 계산 알고리즘

(가) 전열교환기의 성능 DATA 입력

변수명	명칭	단위
현열교환효율	QSHEEFH	%
잠열교환효율	QLHEEFH	%

(나) 환기(RA) 또는 배기(EA)의 조건

(배기 Fan에 의한 온도상승은 무시한다. 즉, 배기 Duct에 의한 온도상승은 환기 Duct 온도상승에 포함되어 있는 것으로 본다.)

RADBH = SYRMDBH

RAXH = SYRMXH

EADBH = RADBH

EAXH = RAXH

(다) 전열교환기 통과후의 공기상태 계산

OADBH2 = OADBH + QSHEEFH × (EADBH-OADBH) / 100

OAXH2 = OAXH + QLHEEFH × (EAXH-OAXH) / 100

OADBH2, OAXH2 로부터 → OARHH2, OAWBH2, OAEPH2 습공기 계산

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
QSHEEFH	현열교환효율	%	100	0	55	
QLHEEFH	잠열교환효율	%	100	0	65	

(나) 출력변수

변수명	명칭	단위	비고
OADBH2, OAXH2, OAWBH2, OARHH2, OAEPH2	전열교환기 통과후의 공기상태	-	

더. 냉방 혼합공기 상태계산(F_MIXC)

(1) 계산 알고리즘

(가) 혼합공기의 건구온도 계산

$$MADBC(N) = \frac{(RADBC(N) \times RAVC) + (OADBC2(N) \times OAVC)}{(RAVC + OAVC)}$$

(나) 혼합공기의 절대습도 계산

$$MAXC(N) = \frac{(RAXC(N) \times RAVC) + (OAXC2(N) \times OAVC)}{(RAVC + OAVC)}$$

(다) MADBC(N), MAXC(N)로 부터 습공기의 상태량 계산

→ MARHC(N), MAWBC(N), MAEPC(N)

(2) 변수명

(가) 내부변수

변수명	명 칭	단 위	비 고
RADBC(N)	환기건구온도	℃	
RAXC(N)	환기절대습도	g/kg	
OADBC2(N)	수정외기건구온도	℃	
OAXC2(N)	수정외기절대습도	g/kg	
OAVC	외기풍량	ℓ/s	
RAVC	환기풍량	ℓ/s	

(나) 출력변수

혼합공기의 상태량

MADBC(N), MAXC(N), MARHC(N), MAWBC(N), MAEPC(N)

러. 난방 혼합공기 상태계산(F_MIXH)

(1) 계산 알고리즘

(가) 혼합공기의 건구온도 계산

$$MADBH = \frac{(RADBH \times RAVH) + (OADB2H \times OAVH)}{(RAVH + OAVH)}$$

(나) 혼합공기의 절대습도 계산

$$MAXH = \frac{(RAXH \times RAVH) + (OAXH2 \times OAVH)}{(RAVH + OAVH)}$$

(다) MADBH, MAXH로부터 습공기 상태량 개산

→ MARHH, MAWBH, MAEPH

(2) 변수명

(가) 내부변수

변수명	명 칭	단 위	비 고
RADBC(N)	환기건구온도	℃	
RAXC(N)	환기절대습도	g/kg	
OADBC2(N)	수정외기건구온도	℃	
OAXC2(N)	수정외기절대습도	g/kg	
OAVC	외기풍량	ℓ/s	
RAVC	환기풍량	ℓ/s	

(나) 출력변수

혼합공기의 상태량

MADBC(N), MAXC(N), MARHC(N), MAWBC(N), MAEPC(N)

머. 난방풍량결정(난방취출온도차, HLADT 기준)(F_SAVH)

(1) 계산 알고리즘

(가) 난방풍량 계산

$$\text{난방풍량 } SAVH = \frac{SYQH}{1.23 \times ACF \times HLADT}$$

(나) 난방코일 출구 급기온도

SADBH = SYRMDBH - HLADT

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
HLADT	난방취출온도차	℃	60	20	40	

(나) 출력변수

변수명	명칭	단위	비고
SADBH	급기건구온도	℃	
SAVH	급기풍량	ℓ/s	
SYQH	시스템난방부하	W	

(다) 내부변수

변수명	명칭	단위	비고
SYRMDBH	대표실내온도조건	℃	

버. 예열코일(외기덕트)의 계산(F_PH)

(1) 계산 알고리즘

(가) 기상자료로부터 겨울 외기상태를 계산 한다.

OADBH, OAXH → OAWBH, OARHH, OAEPH

(나) 예열코일(외기덕트) Data 입력

변수명	명칭	단위	비고
PHCLADB	예열코일출구온도	℃	PHCLADB 의 값은 0 보다 클 것

(다) 만약 OADBH ≥ PHCLADB 이면 예열코일(외기덕트) = (없음)

OADBH2 = OADBH, QPH = 0

또는 OADBH2 = PHCLADB 이면

QPH = OAVH × 1.23 x ACF × (OADBH2 - OADBH)

(라) 예열코일의 선정과 계산

1. 전기

$$PHKW = \frac{QPH}{1000}$$

2. 증기

열원시스템 공조기용 열원의 증기 압력 STP(kPa)로

부터 증기 잠열 STQL(kJ/kg)를 구해 온다.

$$PHKGS(kg/s) = \frac{QPH}{1000 \times STQL}$$

(마) OAXH2 = OAXH

OADBH2, OAXH2 습공기 상태량 계산

→ OARHH2, OAWBH2, OAEPH2

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
PHCLADB	예열코일출구온도	℃	10	0	5	

(나) 출력변수

변수명	명칭	단위	비 고
QPH	예열열량	W	
STQL	증기잠열	(kJ/kg)	
PHKW	예열코일 전기용량	W	
PHKGS	예열코일 증기용량	(kg/s)	
OADBH2, OAXH2, OAWBH2, OARHH2, OAEPH2	외기의 상태량	-	

(다) 내부변수

명칭	변수명	단위	비 고
OADBH OAXH	외기 건구온도 외기 절대습도	℃ g/kg	

서. 예열코일(혼합박스)의 계산(F_PHC)

(1) 계산 알고리즘

(가) 예열코일(외기덕트) Data 입력

변수명	명칭	단위	비고
PHCLADB	예열코일출구온도	℃	PHCLADB 의 값은 0 보다 클 것

(나) 만약 OADBH ≥ PHCLADB 이면 예열코일(외기덕트) = (없음)

OADBH2 = OADBH, QPH = 0

또한 OADBH2 = PHCLADB 이면

QPH = OAVH × 1.23 x ACF × (OADBH2-OADBH)

(다) 예열코일의 선정과 계산

1. 전기

$$PHCELKW = \frac{QPH}{1000}$$

2. 증기

열원시스템 공조기용 열원의 증기 압력 STP(kPa)로

부터 증기 잠열 STQL(kJ/kg)를 구해 온다.

$$PHCSTKGS(kg/s) = \frac{QPH}{1000 \times STQL}$$

(라) MAXH2 = MAXH

MADBH2, MAXH2 습공기 상태량 계산

→ MARHH2, MAWBH2, MAEPH

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
PHCLADB	예열코일출구온도	℃	10	0	5	

(나) 출력변수

변수명	명칭	단위	비 고
QPH	예열열량	W	
STQL	증기잠열	(kJ/kg)	
PHCELKW	예열코일 전기용량	W	
PHCSTKGS	예열코일 증기용량	(kg/s)	
MADBH2, MAXH2, MAWBH2, MARHH2, MAEPH2	혼합공기의 상태량	-	

어. 가습기 계산(F_HUMTR)

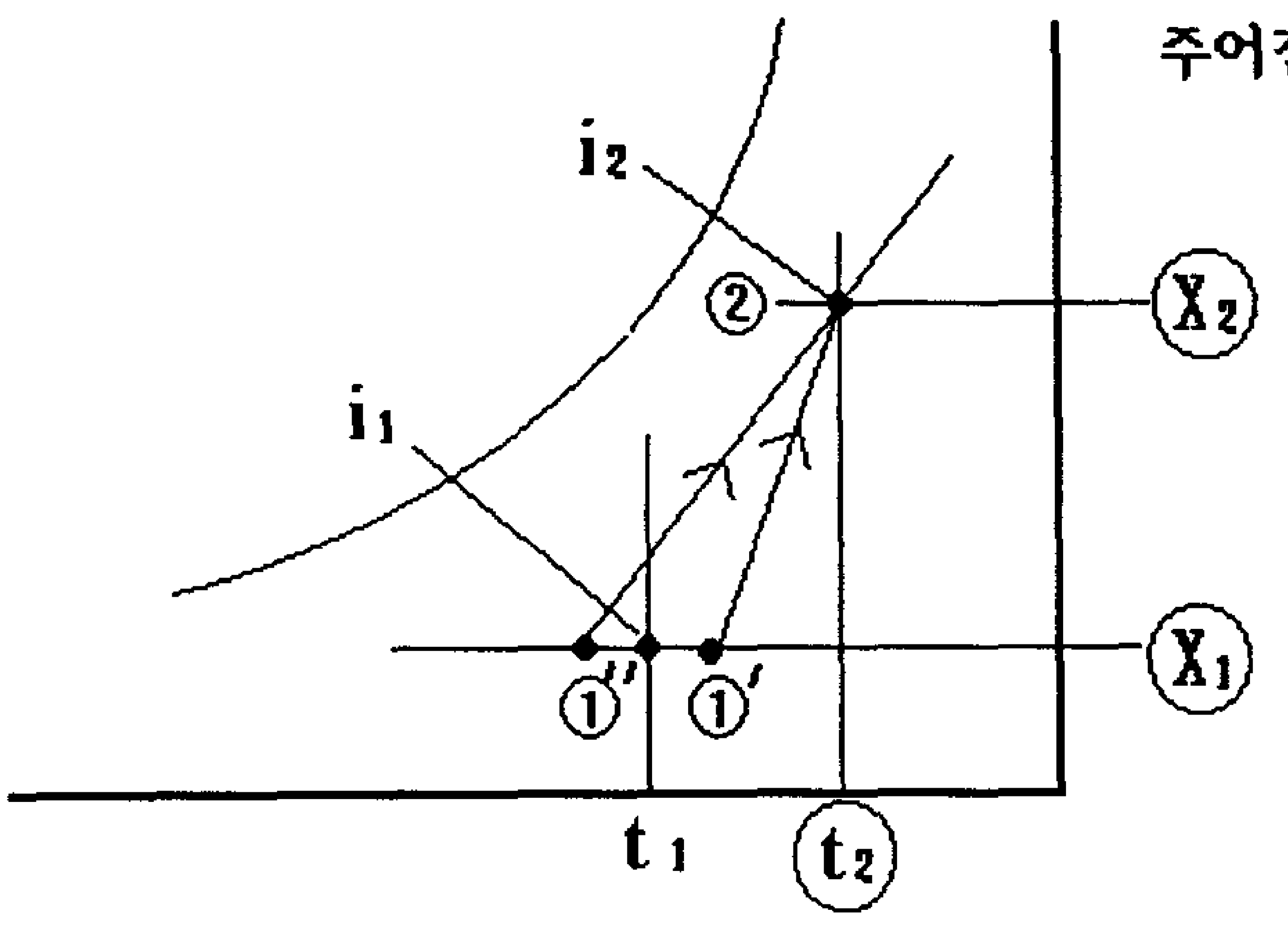
(1) 계산 알고리즘

(가) 가습기의 종류 선택

1. 증기분사식
2. 전기 PAN가습
3. 물분사식
4. 통풍기화식

(나) 증기분사식의 경우

1. 증기분사식의 가습과정



주어진 조건: ②의 상태치 t_2, X_2, i_2
①의 절대습도 X_1

가습량 SYHUMGS = ACSAVH x(X_2 - X_1) x 1.204 x ACF / 3.6 (g/s)

2. 열원시스템의 가습기용 열원 '증기분사' 압력 STP(kPa)을 읽어온다.
해당압력을 찾아 포화증기 엔탈피 STQL(kJ/kg)을 찾아 ①에서의 Enthalpy $\underline{i_1}$을 구한다.

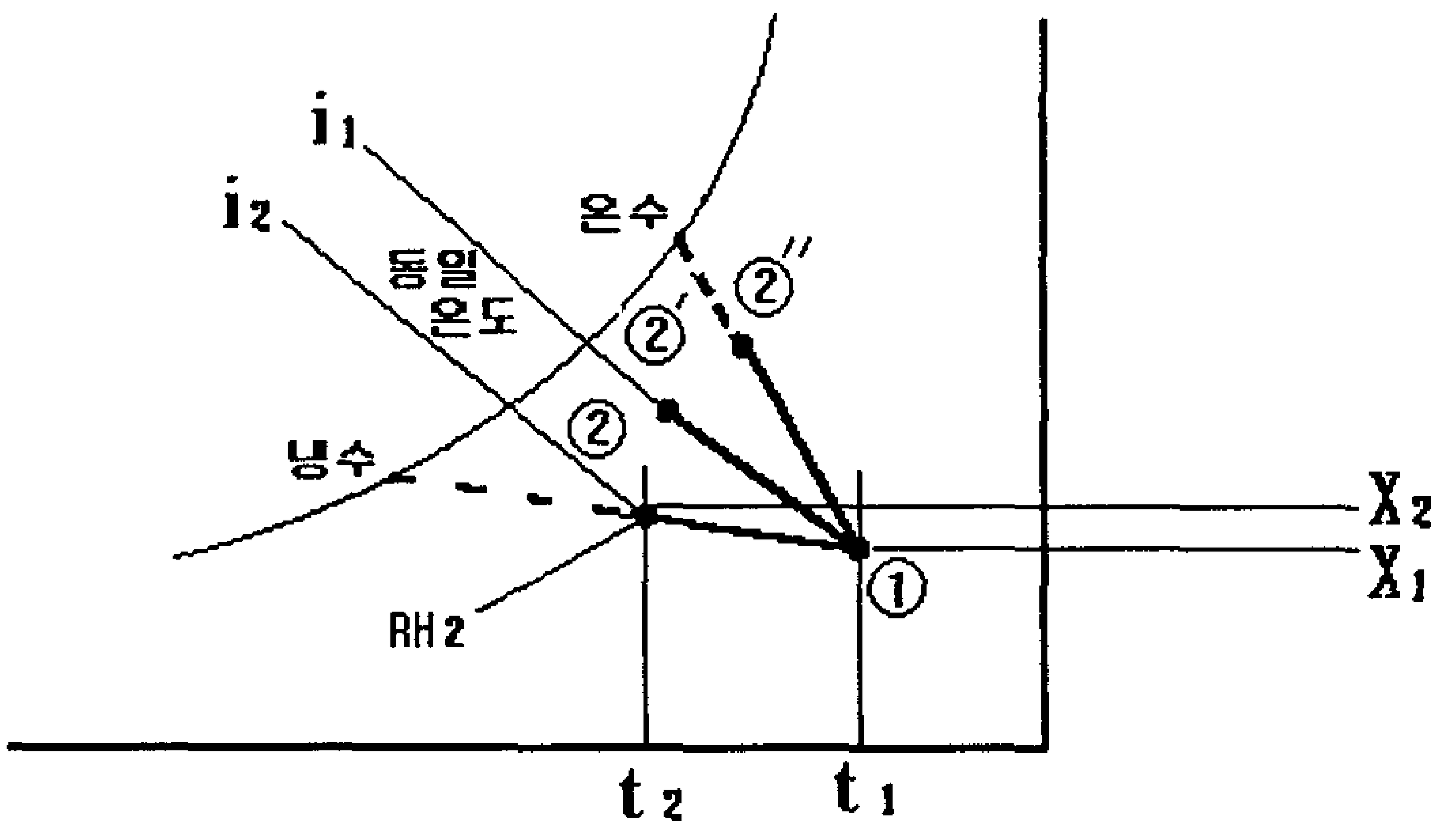
$$i_1 = i_2 - \frac{\text{SYHUMGS x STQL}}{\text{1.204 x ACF x 1000 x ACSAVH}}$$

3. $\underline{i_1}$ 과 $\underline{X_1}$ 으로부터 t_1, t'_1, RH_1 등을 구한다.

증기분사에 의한 온도상승 HUMTR = $\underline{t_2 - t_1}$

(다) 전기PAN가습/물분사식/통풍기화식인 경우

1. 물분무 등의 가습과정



주어진 조건 : ②의 상태치 t_2, X_2, i_2 ····

①의 절대습도 X_1

가습량 SYHUMGS = ACSAVHX(X_2 - X_1) x 1.204 x ACF / 3.6 (g/s)

2. 열원시스템의 가습기용 열원 '전기PAN가습/물분사식/통풍기화식' 수온 HUMTW를 읽어온다.

3. ①에서의 Enthalpy i_1 를 구한다.

$$i_1 = i_2 - \frac{\text{SYHUMGS x HUMTW}}{\text{1.204 x ACF x 1000 x ACSAVH}}$$

4. i_1 과 X_1으로부터 t_1, t'_1, RH_1 등을 구한다.

물가습에 의한 온도상승 HUMTR = $\underline{t_2 - t_1}$ 이다.

(2) 변수명

(가) 입력변수

변수명	명칭	단위	최대	최소	기본값	비 고
STP	증기분사압력	kPa	5	0	0.35	
HUMTW	물/전기/통풍기 화식 수온	℃	100	0	15	

(나) 출력변수

변수명	명칭	단위	비 고
HUMTR	가습에 의한 온도상승	℃	
SYHUMGS	가습량	g/s	

저. 공기밀도 수정계수(F_ACF)

(1) 계산 알고리즘

ACF (Air Density CorreCtion FaCtor) = P / Po

여기서, P = f (Z) : 표고 Z(m) 에 따라 계산된 대기압

P0 = 101.325 kPa

(2) 변수명

(가) 입력변수

명칭	변수명	단위	최대	최소	기본값	비 고
P	표고 Z(m)에 따라 계산된 대기압	kPa	-	-	-	
P0	표준대기압	kPa	-	-	101.325	

(나) 출력변수

명칭	변수명	단위	비 고
ACF	공기밀도 수정계수	-	

5. 공조시스템 계산 알고리즘

가. 단일구역 정풍량 방식(CAV-SZ)

(1) 계산조건의 입력

(가) 풍량결정기준 입력

- 냉방취출온도차(CLADT) 입력
- 터미널 유닛은 없음

(나) 냉각코일 계산조건 입력

1. 냉각코일의 종류 선정

- 냉매코일, · 냉수코일

2. 냉각코일 출구상태(CCSARHMX) 입력

- 냉매코일의 경우 : 기본값 85%
- 냉수코일의 경우 : 기본값 95%

(2) 냉방 계산 알고리즘

(가) ROOM의 부하계산 결과 자료를 읽어온다.

SYBKQS, SYBKQL, SYPLMQ(N)

(나) 대표실내온습도 조건(여름)의 상태량 계산

SYRMDBC, SYRMRHC --> SYRMXC, SYRMWBC, RMEPC

(다) 급기 Fan이 "DRAW-THRU" 방식일 경우의 온도상승 계산

함수 : F_SAFTR

Fan에 의한 온도상승(SAFTR) 계산

이외의 경우 SAFTR = 0, QSAF = 0

(라) Supply Duct의 온도상승 계산

SADTR 입력

(마) 냉방풍량 계산

냉방취출온도차, CLADT 기준 계산 함수 : F_SAVC

(바) 각 ROOM 별 냉방급기풍량수정계산

함수 : F_RMSAVC

(사) 공조기냉방풍량 계산

ACSAVC = Σ(RMSAVC(NR) x RMMT(NR) x ZOMT(NZ))

(아) 외기량 계산

함수 : F_OAVC

(자) 환기량 계산

함수 : F_RAVC

(차) 환기배기 Fan 풍량 계산

함수 : F_RAFVC, F_EAFVC

(카) 실내 → RA로 나가는 풍량계산

함수 : F_RMRAVC

(파) RA Fan 및 RA Duct의 온도상승 계산

함수 : F_RAFTR, F_RADTR

(하) 각 시각(N)에 대해 계산하면서 공조기부하(QCC)가 최대로 되는 냉각코일부하 피크시각(CCQPN)을 찾아낸다.

(거) Plenum 부하에 의한 RA온도상승 계산

함수 : F_RAPLMTR(N)

(너) 외기의 습공기 상태량 계산

기상자료로부터 외기 DATA를 읽어 와서

OADBC(N), OAXC(N) --> OAWBC(N), OARHC(N), OAEPC(N)

(더) 외기부하의 계산

1. 전열교환기가 있는 경우의 계산

- 함수 F_HEOAC를 이용하여 외기 상태량 계산

  OADBC2(N), OAXC2(N), OAWBC2(N), OARHC2(N), OAEPC2(N)

- 외기 부하의 계산

  QCV(N) = OAVC x 1.204 x ACF x (OAEPC2(N) - SYRMEPC)

2. 전열교환기가 없는 경우의 계산

- 외기 부하의 계산

  QCV(N) = OAVC x 1.204 x ACF x (OAEPC(N) - SYRMEPC)

(러) RA와 OA의 혼합상태 계산

MADBC(N), MAXC(N), MARHC(N), MAWBC(N), MAEPC(N)

(머) 급기 FAN이 “Blow-Thru” 방식일 경우의 계산

함수 : F_SAFTR

Fan에 의한 온도상승(SAFTR) 계산

Fan에 의한 열취득계산(QSAF)

(버) 공조기 냉각코일부하 계산

QCC(N) = ACSAVC x 1.204 x ACF x(LCSAEP - MAEPC(N)) + QSAFB

(서) AHU SYSTEM 분석치 계산

(3) 난방계산 알고리즘

(가) 공조기 급기풍량

ACSAVH = ACSAVC

(나) 외기량 계산

함수 : F_OAVH

(다) 기상자료로부터 OADBH, OAXH를 읽어와서 상태량을 계산 한다.

-> OARHH, OAWBH, OAEPH

(라) 공조기 입력화면으로부터 대표실내온습도조건 SYRMDBH, SYRMRHH로 상태량 계산 -> SYRMXH, SYRMWBH, SYRMEPH

(마) 예열코일(외기덕트) 계산

함수 : F_PHC

OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2

(바) 전열교환기 계산

전열교환기 처리후 공기상태량 계산,

OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2

(사) RA와 OA의 혼합상태 계산

RADBH, OADBH2 / RAXH, OAXH2 → MADBH, MAXH → MARHH, MAWBH, MAEPH.

(아) 예열코일(혼합박스) 계산

MADBH2, MAXH2, MARHH2, MAWBH2, MAEPH2

(자) 난방급기온도 계산

1. 실내 난방기기가 있을 때

난방급기온도(SADBH) = SYRMDBH

2. 실내 난방기기가 없을 때

$$\text{난방급기온도(SADBH)} = SYRMDBH + \frac{SYQH}{1.23 \times ACF \times ACSAVH}$$

· 이때 난방급기온도가 55℃이상이면 “Warwing Message”

SAXH = SYRMXH + SYHUMGS / (0.001204 x ACF x ACSAVH)

(차) 가습기 계산

함수 : F_HUMTR

(카) 가열코일 용량 계산

QHC = ACSAVH x 1.23 x ACF x (SADBH - HUMTR - MADBH2)

· 전기일 경우 HCKW = QHC / 1000

· 온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)

· 증기일 경우 HCKGS = QHC / STQL

(파) 각 실별 난방 계산

1. 실내 난방기기가 있을 때

난방기기용량 계산

RMHTG(NR) = RMQH(NR) + RMSAVH(NR) x 1.23
x ACF x (RMDBH(NR) - SADBH)

· 전기일 경우 RMHTGKW(NR) = RMHTG(NR) / 1000

· 온수(1)일 경우 RMHTGLPS(NR) = RMHTG(NR)
x 0.86/(3600 x (HWS1 - HWR1))

· 온수(2)일 경우 RMHTGLPS(NR) = RMHTG(NR)
x 0.86/(3600 x (HWS2 - HWR2))

· 증기일 경우 RMHTGKGS(NR) = RMHTG(NR) / STQL

2. 실내 난방기기가 없을 때

각 실별 난방공급열량

RMQH(NR) = RMSAVH(NR) x 1.23 x ACF x (SADBH - RMDBH(NR))

각 실별 난방과부족량

RMHC(NR) = RMQH(NR) - RMQH(NR)

※ 여기서, RMSAVH = RMSAVC (각 실별풍량)

나. 정풍량 말단재열 방식(CAV-TR)

(1) 계산 조건의 입력

(가) 터미널 유닛(T.U)의 형식

CAV Box, CAV Box with Heating Coil

(나) 풍량결정기준

냉방취출온도차(CLADT) (℃)

(2) 냉방 계산 알고리즘

(가) ~ (버) : 가. 단일구역 정풍량방식의 (2) 냉방 계산 알고리즘의 (가) ~ (버)와 동일

(서) 재열부하 계산(QRH)

QRH = ACSAVC x 1.23 x ACF x (SYRMDBH - SADBC)

(저) AHU SYSTEM 분석치 계산

(3) 터미널유니트(TU_CAV-TR)냉방 계산 알고리즘

(가) 사용할 수 있는 Terminal Unit 종류

CAV Box, CAV Box with Heating Coil

(나) 존의 급기풍량 계산

$$ZOSAVC(NZ) = \sum_{NR(소속zone)} (RMSAVC(NR) \times RMMT(NR))$$

(다) 재열 용량 계산

1. 터미널 유닛을 CAV Box with Heating Coil로 선택한 Zone의 경우

ZOQRH(NZ) = ZOSAVC(NZ) x 1.23
x ACF x (ZORMDBH(NZ) - SADBC)

RHKW(NZ) = ZOQRH(NZ) / 1000 (전기일 경우)

RHLPS1(NZ) = ZORH(NZ)
x 0.86/((3600 x (HWS1 - HWR1)) (온수1일 경우)

RHLPS2(NZ) = ZOQRH(NZ)

x 0.86/((3600 x (HWS2 - HWR2)) (온수2일 경우)

2. 터미널 유닛을 CAV Box로 선택한 Zone의 경우

ZOQRH(NZ) = 0

RHKW(NZ) = 0 (전기일 경우)

RHLPS1,2(NZ) = 0 (온수1,2일 경우)

(4) 난방 계산 알고리즘

(가) ~ (차) : 가. 단일구역 정풍량방식의 (3) 난방 계산 알고리즘의 (가) ~ (차)와 동일

(카) 가열코일 용량 계산

1. 만약 MADBH2 > (SADBH - HUMTR) 이면

SADBH = MADBH2 + HUMTR

QHC = 0, HCKW = 0

HCLPS = 0, HCKGS = 0

2. else QHC = ACSAVH x 1.23 x ACF

x (SADBH - HUMTR - MADBH2)

- 전기일 경우 HCKW = QHC / 1000
- 온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)
- 증기일 경우 HCKGS = QHC / STQL

(5) 터미널유닛(TU_CAV-TR)의 난방계산 알고리즘

(가) 존의 급기풍량 계산

ZOSAVH(NZ) = ZOSAVC(NZ)

(나) 존의 급기온도와 가열 열량 계산

1. 터미널 유닛을 CAV Box with Heating Coil로 선택한 Zone

ZOSADBH(NZ) = ZOQH(NZ)/(ZOSAVH(NZ)

x 1.23 x ACF) + ZORMDBH(NZ)

ZOQHC(NZ) = ZOQH(NZ) + ZOQRH(NZ)

2. 터미널 유닛을 CAV Box로 선택한 Zone

ZOSADBH(NZ)= SADBH, ZOQHC(NZ) = 0

(다) 각 실별 난방계산(의미없음)

1. 실내난방공급량 계산

RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF

x (ZOSADBH(NZ) - RMDBH(NR))

※ 여기서, RMSAVH(NR) = RMSAVC(NR) (각 실별풍량)

2. 실내난방과부족 계산

RMHC(NR) = RMQHS(NR) - RMQH(NR)

다. 정풍량 멀티존/이중덕트 방식(CAV-MZ/DD)

(1) 계산조건

(가) MZ의 경우

T.U는 없음. 그러나 ZONE은 있음

(나) DD의 경우

T.U는 CAV Box(Cooling only), CAV Mixing DD Box

(다) 풍량결정기준

냉풍덕트온도(TCD) (℃)

온풍덕트온도(THD) (℃)

(2) 냉방계산 알고리즘

(가) ~ (라) : 가. 단일구역 정풍량방식의 (2) 냉방계산 알고리즘의 (가) ~ (라)와 동일

(마) 냉방풍량 계산

SADBC = TCD로 하여

냉방풍량 (SAVC) 계산---(냉풍덕트온도, SADBC)

(바) ~ (머) : 가. 단일구역 정풍량방식의 (1) 냉방계산 알고리즘의 (바) ~ (머)와 동일

(버) 냉풍 및 온풍 덕트의 풍량 계산

$$ACSAVC_CD = \frac{SYBKQS + ACSAVC \times 1.23 \times ACF \times (MADBC(N) + SAFTRD + SADTR - SYRMDBC)}{1.23 \times ACF \times (MADBC(N) + SAFTRD - CCLADB)}$$

ACSAVC_HD = ACSAVC - ACSAVC_CD

(서) AHU SYSTEM 분석치 계산

(3) 터미널유닛(TU_CAV_MZ/DD)의 냉방계산 알고리즘

(가) 존의 급기풍량 계산

$$ZOSAVC(NZ)=\Sigma(RMSAVC(NR) \times RMMT(NR))$$

(나) 터미널 유닛을 CAV Box DD Mixing으로 선택한 Zone의 계산

ZOQRH(NZ) = 0, RHKW(NZ) = 0, RHLPS(NZ) = 0

Zone Level 에서의 여름철CD, HD 풍량계산

$$ZOSAVC_CD(NZ) = \frac{ZOBKQS(NZ) + ZOSAVC(NZ) \times 1.23 \times ACF \times (MADBC(N) + SAFTRD + SADTR - ZORMDBC(NZ))}{1.23 \times ACF \times (MADBC(N) + SAFTRD - CCLADB)}$$

$$ZOSAVC_HD(NZ) = ZOSAVC(NZ) - ZOSAVC_CD(NZ)$$

(다) 터미널 유닛을 CAV Box로 선택한 Zone의 계산

$$ZOSAVC_CD(NZ) = ZOSAVC(NZ)$$

$$ZOSAVC_HD(NZ) = 0$$

(4) 난방계산 알고리즘

(가) ~ (아) : 가. 단일구역 정풍량방식의 (3) 난방계산 알고리즘의 (가) ~ (아)와 동일

(자) 급기온도 및 절대습도 계산

$$SADBH = MADBH2$$

$$SAXH = SYRMXH + SYHUMGS / (0.001204 \times ACF \times ACSAVH)$$

(차) 가습기 계산

함수 : F_HUMTR

(카) 냉풍덕트와 온풍덕트의 풍량 계산

$$ACSAVH_HD = \frac{SYQH + ACSAVH \times 1.23 \times ACF \times (SYRMDBH - MADBH2 - HUMTR)}{1.23 \times ACF \times (THD - MADBH2 - HUMTR)}$$

$$ACSAVH_CD = ACSAVH - ACSAVH_HD$$

(파) 가열코일용량 계산

$$QHC = ACSAVH_HD \times 1.23 \times ACF \times (THD - MADBH2 - HUMTR)$$

(5) 터미널 유닛(TU_CAV-MZ/DD) 난방계산 알고리즘

(가) 존의 급기풍량 계산

ZOSAVH(NZ) = ZOSAVC(NZ)

1. 터미널 유닛을 CAV Box DD Mixing로 선택한 Zone의 경우

ZOSADBH(NZ) = ZOQH(NZ)

/ (ZOSAVH(NZ) x 1.23 x ACF) + ZORMDBH(NZ)

ZOQHC(NZ) = 0

Zone Level에서의 겨울철 CD, HD 풍량계산

$$ZOSAVH_HD(NZ) = \frac{ZOQH(NZ) + ZOSAVH(NZ) \times 1.23 \times ACF \times (ZORMDBH(NZ) - MADBH2 - HUMTR)}{1.23 \times ACF \times (THD - MADBH2 - HUMTR)}$$

ZOSAVH_CD(NZ) = ZOSAVH(NZ) - ZOSAVH_HD(NZ)

2. 터미널 유닛을 CAV Box로 선택한 Zone의 경우

ZOSADBH(NZ) = MADBH2 + HUMTR

ZOQHC(NZ) = 0

Zone Level에서의 겨울철 CD, HD 풍량계산

ZOSAVH_HD(NZ) = 0

ZOSAVH_CD(NZ) = ZOSAVH(NZ)

(나) 각 실별 난방계산 (의미없음)

1. 각 실별 난방공급열량

RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF

x (ZOSADBH(NZ) - RMDBH(NR))

여기서, RMSAVH = RMSAVC (각 실별풍량)

2. 각 실별 난방 과부족량

RMHC(NR) = RMQHS(NR) - RMQH(NR)

라. 정풍량 유인유닛/팬코일 유닛 방식(CAV-ID / CAV-FC)

(1) 계산조건의 입력

(가) 풍량결정 기준

- 여름풍량 = 여름외기량(ACSAVC = OAVC)

· 터미널 유닛은 없음

1. 외기량결정기준

각 Room 에 대해 여름외기량 산정기준 입력으로 부터 각각의 외기량 계산기준에 따라 계산하고, 최대 또는 최소를 선택한다. (=RMOAVC)

OAVPPC	(ℓ/s)/인 × RMPP	max or min
OAVARC	(ℓ/s)/m² × RMAR	
OAXNGC	x/Hr × RMVOL / 3.6	
OASAVC	(l/s) 직접입력값	

2. 실내배기량과 비교

만약 RMSAEA(NR) = "y" 이고 RMOAVC(NR) < RMEAV(NR),

이면 RMOAVC(NR) = RMEAV(NR)

3. 실별 최소급기풍량과 비교

만약 RMOAVC(NR) < RMSAVMI(NR) 이면

RMOAVC(NR) = RMSAVMI(NR)

4. RMSAVC(NR) = RMOAVC(NR)

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일, * 냉수코일

2. 냉각코일 출구상태(CCSARHMX) 입력

· 냉매코일의 경우 : 기본값 값 85%

· 냉수코일의 경우 : 기본값 값 95%

(2) 냉방계산 알고리즘

(가) ~ (라) : 가. 단일구역 정풍량방식의 (2) 냉방계산 알고리즘의 (가) ~ (라)와 동일

(마) 냉각코일 출구 공기의 상태량 계산

CCLADBC = SADBC - (SAFTRD + SADTR)

CCLADBC, CCLARHC -> CCLAWBC, CCLAXC, CCLAEPC

(바) 급기의 상태량 계산

SAXC = CCLAXC

SADBC, SAXC --> SAWBC, SARHC, SAEPC

(사) 공조기냉방풍량

ACSAVC = Σ(RMSAVC(NR) x RMMT(NR) x ZOMT(NZ))

(아) 환기량 계산

함수 : F_RAVC

(자) 배기Fan 풍량 계산

함수 : F_EAFVC

(차) 실내 → RA로 나가는 풍량계산

함수 : F_RMRAVC

(카) RA Fan 및 RA Duct의 온도상승계산

함수 : F_RAFTR, F_RADTR

(파) 각 시각(N)에 대해 계산하면서 공조기부하(QCC)가 최대로 되는 냉각코일부하 피크시각(CCQPN)을 찾아낸다.

(하) Plenum 부하에 의한 RA온도상승 계산

함수 : F_RAPLMTR(N)

(거) 외기의 습공기 상태량 계산

OADBC(N), OAXC(N)

-> OAWBC(N), OARHC(N), OAEPC(N)

(너) 전열교환기 계산

OADBC2(N), OAXC2(N), OAWBC2(N), OARHC2(N),OAEPC2(N)

(더) 공조기 냉각코일부하 계산

1. 전열교환기가 있는 경우

QCC = ACSAVC x 1.204 x ACF x (OAEP2(N) - CCLAEP) + QSAFB

2. 전열교환기가 없는 경우

QCC = ACSAVC x 1.204 x ACF x (OAEPC(N) - CCLAEP) + QSAFB

(러) AHU SYSTEM 분석치 계산

(3) 각 실별 유인 유닛/팬코일 유닛(IDU /FCU)의 용량계산

(가) ROOM의 부하계산 결과 자료를 읽어온다.

RMQPS(NR), RMQPLM(NR),

(나) 유닛의 실별부하 집계

IDUQS(NR) = RMQPS(NR) - RMSAVC(NR)

x 1.23 x ACF x (RMDBC(NR) - SADBC)

IDUQL(NR) = RMQPLM(NR) - RMSAVC(NR)

x 3.01 x ACF x (RMXC(NR) - SAXC)

IDUQT(NR) = IDUQS(NR) + IDUQL(NR)

(4) 난방계산 알고리즘

(가) 외기량결정기준

1. 겨울외기량 산정기준 입력으로 부터 각각의 외기량 계산기준에 따라 계산하고, 최대 또는 최소를 선택한다. (=RMOAVH)

OAVPPH	(ℓ/s)/인 × RMPP	max or min
OAVARH	(ℓ/s)/m^2 × RMAR	
OAXNGH	x/Hr × RMVOL / 3.6	
OASAVH	(l/s) 직접입력값	

2. 실내배기량과 비교

만약 RMSAEA(NR) = "y" 이고 RMOAVH(NR) < RMEAV(NR) 이면

RMOAVH(NR) = RMEAV(NR)

3. 실별 최소급기풍량과 비교

만약 RMOAVH(NR) < RMSAVMI(NR) 이면

RMOAVH(NR) = RMSAVMI(NR)

4. RMSAVH(NR) = RMOAVH(NR)

(나) 공조기 난방풍량 계산

ACSAVH = Σ(RMSAVH(NR) x RMMT(NR) x ZOMT(NZ))

(다) 기상자료로부터 OADBH, OAXH를 읽어와서 습공기 상태량 계산

OARHH, OAWBH, OAEPH

(라) 예열코일(외기덕트) 계산

예열코일계산, OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2 구함.

(마) 전열교환기 계산

전열교환기 처리후 공기상태를 계산

OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2

(바) 대표실내온습도 조건(여름)의 상태량 계산

SYRMDBH, SYRMRHH --> SYRMXH, SYRMWBH, SYRMEPH

(사) 급기의 상태량 계산

SADBH = SYRMDBH

SAXH = SYRMXH + SYHUMGS / (0.001204 x ACF x ACSAVH)

SADBC, SAXC --> SAWBC, SARHC, SAEPC

(아) 가습기 계산

함수 : F_HUMTR

(자) 가열코일 용량 계산

QHC = ACSAVH x 1.23 x ACF x (SADBH - HUMTR - OADBH2)

전기일 경우 HCKW = QHC / 1000

온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)

증기일 경우 HCKGS = QHC / STQL

(차) 각 실별 IDU 용량 계산

IDUQH(NR) = RMQH(NR) - RMSAVH(NR)

x 1.23 x ACF x (SADBH - RMDBH(NR))

마. 변풍량 외주부 공기난방 방식(VAV-Rad)

(1) 계산조건의 입력

(가) 풍량결정기준

· 냉방취출온도차(CLADT) 입력

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일, · 냉수코일

2. 냉각코일 출구상태(CCLARHMX) 입력

· 냉매코일의 경우 : 기본값 85%

· 냉수코일의 경우 : 기본값 95%

(다) 터미널 유닛의 종류 :

VAV Box, CAV Box, CAV w/ Heating Coil

(2) 냉방계산 알고리즘

(가) ROOM의 부하계산 결과 자료를 읽어온다.

SYBKQS, SYBKQL, SYPLMQ(N)

(VAV System에 있어서 SYBKQS, SYBKQL을 구할 때 T.U의 선택이 CAV Box 또는 CAV Box w/ Htg Coil인 Zone에 대해서는 RMQPS, RMQPLM을 적산하고 VAV Box인 Zone에 대해서는 RMQS 값을 적산한 후 최대로 되는 시각의 RMQS적산치, RMQL적산치를 각각 합하여 최종 RMBKQS 및 RMBKQL을 계산한다는 점에 특히 유의할 것)

(나) ~ (마) : 가. 단일구역 정풍량방식의 (2) 냉방계산 알고리즘의 (나) ~ (마)와 동일

(바) 각 ROOM 별 풍량 수정계산

함수 : F_RMSAV

1. 터미널 유닛을 CAV Box, CAV Box w/ Htg. Coil을 선택한 Zone에 속하는 각 실에 대해 각 ROOM별 냉방급기풍량(RMSAVC)수정계산

RMSAVD(NR) = RMQPS(NR) / (1.23 x ACF x CLADT)

만약 RMSAVC(NR) < RMSAVMI(NR) 이면

RMSAVC(NR) = RMSAVMI(NR)

만약 RMSAEA(NR) = "y" 이고 RMSAVC(NR) < RMEAV(NR) 이면

RMSAVC(NR) = RMEAV(NR)

ACSAVNEW(NR) = RMSAVC(NR) - RMSAV(NR)

2. 터미널 유닛을 VAV Box로 선택한 Zone에 있어서는 각 실별 최소풍량 설정 또는 급기풍량을 실내배기량 이상으로 지정하여도 의미가 없음.

ACSAVNEW(NR) = 0

(사) 터미널유니트(TU_ VAV-Rad)냉방풍량계산

사용할 수 있는 Terminal Unit 종류 :

VAV Box, CAV Box, CAV Box with Heating Coil

$$ZOSAVC(NZ) = \sum_{NR(\text{소속zone})} (RMSAVC(NR) \times RMMT(NR))$$

(아) 공조기냉방풍량 계산

ACSAVC = SAVC + Σ(ACSAVNEW(NR) x RMMT(NR) x ZOMT(NZ))

(자) 외기량 계산

함수 : F_OAVC

(차) 각 시각(N)에 대해 계산하면서 공조기부하(QCC)가 최대로 되는 냉각코일부하 피크시각(CCQPN)을 찾아낸다.

1. VAV Box로 선택된 Zone에 대해서

SYVAVQS(N) = Σ (RMQS(NR,N) x RMMT(NR) x ZOMT(NZ))

VAVACVC(N) = SYVAVQS(N) / (1.23 x ACF x CLADT)

2. CAV Box로 선택된 Zone에 대해서

CAVACVC = Σ (ZOSAVC(NZ) x RMMT(NR) x ZOMT(NZ))

ACVNEW(N) = VAVACVC(N) + CAVACVC

(카) 외기량 재계산

OAVNEW(N) = OAVC x ACVNEW(N)/ACSAVC

(파) ~ (처) : 가. 단일구역 정풍량방식의 (2) 냉방계산 알고리즘의 (자) ~ (버)와 동일

(3) 난방계산 알고리즘

(가) 터미널 유닛(TU_VAV-Rad)의 난방풍량 계산

1. T.U를 CAV Box, CAV Box w/ Htg. Coil을 선택한 Zone

ZOSAVH(NZ) = ZOSAVC(NZ)

2. T.U를 VAV Box로 선택한 Zone

터미널유니트 자료의 입력 화면에서 최소풍량 설정(난방시 고려)값으로부터 ZOSAVH(NR) =

- 환기회수 ZOSAMIXNG(NZ) x ZOV(NZ) / 3.6
- 면적당 풍량 ZOSAMIAR(NZ) x ZOAR(NZ)
- % of T.U ZOSAMI%TU(NZ) x ZOSAVC(NZ) / 100

(나) 공조기 급기풍량 계산

ACSAVH = Σ(ZOSAVH(NZ) x ZOMT(NZ))

(다) 외기량 계산

OAVH2 = OAVC x ACSAVH / ACSAVC

만약 OAVH > OAVH2 이면 OAVH = OAVH2

주) 겨울의 외기량은 여름외기량으로부터 풍량비율에 따라 계산됨

(라) ~ (카) : 가. 단일구역 정풍량방식의 (3) 난방계산 알고리즘의
(다) ~ (차)와 동일

(파) 가열코일 용량 계산

만약 MADBH2 > (SADBH - HUMTR) 이면 SADBH = MADBH2 + HUMTR

QHC = 0, HCKW = 0, HCLPS = 0, HCKGS = 0

또는, 가열코일용량 QHC = ACSAVH x 1.23 x ACF
x (SADBH - HUMTR - MADBH2)

전기일 경우 HCKW = QHC / 1000

온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)

증기일 경우 HCKGS = QHC / STQL

* (Warm-up) QHCWARMUP = ACSAVC x 1.23 x ACF x (WARMUPHCTR)

전기일 경우 HCKWWARMUP = QHCWARMUP / 1000

온수일 경우 HCLPSWARMUP = QHCWARMUP x 0.86/(3600 x (HWS - WR)

증기일 경우 HCKGSWARMUP = QHCWARMUP / STQL

(4) 터미널유니트(TU_VAV-Rad) 난방계산 알고리즘

(가) 존의 급기풍량 계산

ZOSAVH(NZ), ZOQRHH(NZ) = 0

(나) 존의 급기온도와 가열열량 계산

1. CAV Box w/ Htg Coil의 경우

ZOQRHH(NZ) = ZOSAVH(NZ) x 1.23 x ACF x (ZORMDBH(NZ) - SADBH)

ZOSADBH(NZ)= ZORMDBH(NZ) + ZOQH(NZ)/(ZOSAVH(NZ) x 1.23 x ACF)

ZOQHC(NZ) = ZOQRHH(NZ) + ZOQH(NZ)

전기일 경우 ZOHCKW(NZ) = ZOQHC / 1000

온수1일 경우 ZOHCLPS(NZ)
= ZOQHC(NZ) x 0.86 / (3600 x (HWS1 - HWR1)

온수2일 경우 ZOHCLPS(NZ)
= ZOQHC(NZ) x 0.86 / (3600 x (HWS2 - HWR2)

2. CAV Box의 경우

ZOSADBH(NZ) = SADBH, ZOQHC(NZ) = 0

(다) 각 실별 난방계산(실내 난방기기가 있슴)

난방기기용량(RMHTG(NR)) = RMQH(NR)

전기일 경우 RMHTGKW(NR) = RMHTG(NR) / 1000

온수(1)일 경우 RMHTGLPS(NR) = RMHTG(NR) x 0.86 /(3600 x (HWS1 - HWR1))

온수(2)일 경우 RMHTGLPS(NR)=RMHTG(NR) x 0.86 /(3600 x (HWS2 - HWR2))

증기일 경우 RMHTGKGS(NR) = RMHTG(NR) / STQL

바. 변풍량 단말 재열방식(VAV-TR)

(1) 계산조건의 입력

(가) 풍량결정기준

· 냉방취출온도차(CLADT) 입력

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일, · 냉수코일

2. 냉각코일 출구상태(CCLASARHMX) 입력

· 냉매코일의 경우 : 기본값 85%

· 냉수코일의 경우 : 기본값 95%

(다) 터미널 유닛의 종류

VAV Box, VAV Box w/ Heating Coil, CAV Box w/ Heating Coil

(2) 냉방계산 알고리즘

(가) ~ (바) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의 (2) 냉방계산 알고리즘의 (가) ~ (바)와 동일

(사) 터미널유니트(TU_ VAV-TR)냉방풍량계산

사용할 수 있는 Terminal Unit 종류 :

VAV Box, VAV Box with Heating Coil

CAV Box with Heating Coil

ZOSAVC(NZ) = $\sum_{NR(소속zone)}$ (RMSAVC(NR) x RMMT(NR))

(아) 터미널유니트(TU_VAV-TR) 최소풍량(=난방풍량)계산

1. T.U를 CAV Box w/ Htg. Coil로 선택한 Zone

   ZOSAVH(NZ) = ZOSAVC(NZ)

2. T.U를 VAV Box, VAV Box w/ Htg. Coil로 선택한 Zone

   자료입력화면에서 최소풍량 설정(난방시 고려)값으로부터

   ZOSAVH(NR) =

   - 환기회수 ZOSAMIXNG(NZ) x ZOVOL(NZ) / 3.6
   - 면적당 풍량 ZOSAMIAR(NZ) x ZOAR(NZ)
   - % of T.U ZOSAMI%TU(NZ) x ZOSAVC(NZ) / 100
   - 난방취출온도차 ZOHLADT(NZ)로 계산

     ZOQH(NZ) / (1.23 x ACF x ZOHLADT(NZ))

(자) 공조기냉방풍량 계산

ACSAVC = SAVC + Σ(ACSAVNEW(NR) x RMMT(NR) x ZOMT(NZ))

(차) 외기량(OAVC) 계산

(카) 각 시각(N)에 대해 계산하면서 공조기부하(QCC)가 최대로 되는 냉각코일부하 피크시각(CCQPN)을 찾아낸다.

1. VAV Box로 선택된 Zone에 대해서

   SYVAVQS(N) = Σ (RMQS(NR,N) x RMMT(NR) x ZOMT(NZ))

   VAVACVC(N) = SYVAVQS(N) / (1.23 x ACF x CLADT)

2. CAV Box로 선택된 Zone에 대해서

   CAVACVC = Σ (ZOSAVC(NZ) x RMMT(NR) x ZOMT(NZ))

   ACVNEW(N) = VAVACVC(N) + CAVACVC

(파) 외기량 재계산

OAVNEW(N) = OAVC x ACVNEW(N)/ACSAVC

(하) ~ (처) : 가. 단일구역 정풍량방식의 (2) 냉방계산 알고리즘의 (자) ~ (버)와 동일

(3) 난방계산 알고리즘

(가) ~ (카) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의 (3) 난방 방식 알고리즘의 (나) ~ (파)와 동일

(4) 터미널유니트(TU_VAV-TR) 난방계산 알고리즘

(가) 존의 급기풍량

ZOSAVH(NZ)

(나) 존의 급기온도와 급기풍량 계산

1. 터미널 유닛을 VAV Box로 선택한 Zone

ZOQRHH(NZ) = 0, ZOSADBH(NZ) = SADBH, ZOQHC = 0

ZOQRHH(NZ) = ZOSAVH(NZ) x 1.23 x ACF x (ZORMDBH(NZ) - SADBC)

ZOSADBH(NZ) = ZORMDBH(NZ) + ZOQH(NZ)/(ZOSAVH(NZ) x 1.23 x ACF)

ZOQHC(NZ) = ZOQRHH(NZ) + ZOQH(NZ)

전기일 경우 ZOHCKW(NZ) = ZOQHC / 1000

온수1일 경우 ZOHCLPS(NZ)
= ZOQHC(NZ) x 0.86 / (3600 x (HWS1 - HWR1)

온수2일 경우 ZOHCLPS(NZ)
= ZOQHC(NZ) x 0.86 / (3600 x (HWS2 - HWR2)

(다) 각 실별 난방계산(의미없음)

실별난방공급량 RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF
x (ZOSADBH(NZ) - RMDBH(NR))

실별난방과부족 RMHC(NR) = RMQHS(NR) - RMQH(NR)

사. 변풍량 병렬 팬급기 방식(VAV-FPPL)

(1) 계산조건의 입력

(가) 풍량결정기준

· 냉방취출온도차(CLADT) 입력

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일, · 냉수코일

2. 냉각코일 출구상태(CCSARHMX) 입력

- 냉매코일의 경우 : 기본값 85%
- 냉수코일의 경우 : 기본값 95%

(다) 터미널 유닛의 종류 ;

VAV Box, VAV Box Fan Powered (Parallel),

CAV Box w/ Heating Coil

(2) 냉방계산 알고리즘

(가) ~ (바) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의
(2) 냉방계산 알고리즘의 (가) ~ (바)와 동일

(사) 터미널유니트(TU_ VAV-FPPL)냉방풍량계산

사용할 수 있는 Terminal Unit 종류 ;

VAV Box, VAV Box Fan Powered (Parallel),

CAV Box with Heating Coil

$$ZOSAVC(NZ) = \sum_{NR(소속zone)} (RMSAVC(NR) \times RMMT(NR))$$

(아) 터미널유니트(TU_VAV-FPPL) 최소풍량계산(난방풍량과는 틀림)

1. 터미널 유닛을 CAV Box w/ Htg. Coil로 선택한 Zone

   ZOSAVMI(NZ) = ZOSAVC(NZ)

2. 터미널 유닛을 VAV Box, VAV Box FPPL로 선택한Zone

   자료입력화면에서 최소풍량 설정(난방시 고려)값으로부터

   ZOSAVMI(NZ) =

   - 환기회수 = ZOSAMIXNG(NZ) x ZOV(NZ) / 3.6
   - 면적당 풍량 = ZOSAMIAR(NZ) x ZOAR(NZ)
   - % of T.U = ZOSAMI%TU(NZ) x ZOSAVC(NZ) / 100

   ZOQRH(NZ) = ZOSAVMI(NZ) x 1.23 x ACF
   x (ZORMDBH(NZ) - SADBC)

(자) ~ (처) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의
(2) 냉방계산 알고리즘의 (자) ~ (처)와 동일

(3) 난방계산 알고리즘

(가) ~ (카) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의 (3) 난방

방식 알고리즘의 (나) ~ (파)와 동일

(4) 터미널 유닛(TU_VAV-FPPL)의 난방계산 알고리즘

(가) 터미널 유닛을 VAV Box, CAV Box w/ Htg. Coil로 선택한 Zone

ZOSAVH(NZ) = ZOSAVMI(NZ)

(나) 터미널 유닛을 VAV Box Fan Powered (Parallel)로 선택한 Zone

자료입력화면에서 Fan용량 설정(난방시)값으로부터

ZOFV(NZ) =

- 환기회수 ZOSAMIXNG(NZ) x ZOV(NZ) / 3.6
- 면적당 풍량 ZOSAMIAR(NZ) x ZOAR(NZ)
- % of T.U ZOSAMI%TU(NZ) x ZOSAVC(NZ) / 100
- 난방취출온도차 ZOHLADT(NZ)로 계산

ZOQH(NZ)/(1.23 x ACF x ZOHLADT(NZ)) - ZOSAVMI(NZ)

ZOSAVH(NZ) = ZOFV(NZ) + ZOSAVMI(NZ)

ZOQRHH(NZ) = ZOSAVMI(NZ) x 1.23 x ACF x (ZORMDBH(NZ) - SADBC)

ZOQPLMTRH(NZ) = Σ(RMLIGHTPLM x RMMT) /(1.23 x ACF x ZOFV(NZ))

ZOQPLM(NZ) = ZOFV(NZ) x 1.23 x ACF x ZOQPLMTRH(NZ)

(다) 터미널 유닛을 CAV Box w/ Htg. Coil로 선택한 Zone

ZOQPLM(NZ) = 0

ZOSADBH(NZ) = ZORMDBH(NZ) + ZOQH(NZ)/(ZOSAVH(NZ) x 1.23 x ACF)

ZOQHC(NZ) = ZOQRHH(NZ) + ZOQH(NZ) - ZOQPLM(NZ)

전기일 경우 ZOHCKW(NZ) = ZOQHC/1000

온수1일 경우 ZOHCLPS(NZ) = ZOQHC(NZ) x 0.86/(3600 x (HWS1 - HWR1)

온수2일 경우 ZOHCLPS(NZ) = ZOQHC(NZ) x 0.86/(3600 x (HWS2 - HWR2)

(라) 각 실별 난방계산(의미없음)

실별난방공급량 RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF

x (ZOSADBH(NZ) - RMDBH(NR))

실별난방과부족 RMHC(NR) = RMQHS(NR) - RMQH(NR)

아. 변풍량 직렬 팬급기 방식(VAV-FPSR)

(1) 계산조건의 입력

(가) 풍량결정기준

· 냉방취출온도차(CLADT) 입력

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일, · 냉수코일

2. 냉각코일 출구상태(CCSARHMX) 입력

· 냉매코일의 경우 : 기본값 85%

· 냉수코일의 경우 : 기본값 95%

(다) 터미널 유닛의 종류 ;

VAV Box, VAV Box Fan Powered (Series)

CAV Box w/ Heating Coil

(2) 냉방계산 알고리즘

(가) ~ (바) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의 (2) 냉방계산 알고리즘의 (가) ~ (바)와 동일

(사) 터미널유니트(TU_ VAV-FPSR)냉방풍량계산

사용할 수 있는 Terminal Unit 종류 ;

VAV Box, VAV Box Fan Powered (Series),

CAV Box with Heating Coil

$$ZOSAVC(NZ) = \sum_{NR(소속zone)} (RMSAVC(NR) \times RMMT(NR))$$

(아) 터미널유니트(TU_VAV-FPSR) 최소풍량계산(난방풍량과는 틀림)

1. 터미널 유닛을 CAV Box w/ Htg. Coil로 선택한 Zone

ZOSAVMI(NZ) = ZOSAVC(NZ)

2. 터미널 유닛을 VAV Box, VAV Box FPSR로 선택한Zone

자료입력화면에서 최소풍량 설정(난방시 고려)값으로부터

ZOSAVMI(NZ)

· 환기회수 = ZOSAMIXNG(NZ) x ZOV(NZ) / 3.6

· 면적당 풍량 = ZOSAMIXNG(NZ) x ZOAR(NZ)

· % of T.U = ZOSAMI%TU(NZ) x ZOSAVC(NZ) / 100

ZOQRH(NZ) = ZOSAVMI(NZ) x 1.23 x ACF x (ZORMDBH(NZ) - SADBC)

(자) ~ (처) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의 (2) 냉방계산 알고리즘의 (자) ~ (처)와 동일

(3) 난방계산 알고리즘

(가) ~ (카) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의 (3) 난방 방식 알고리즘의 (나) ~ (파)와 동일

(4) 터미널 유닛(TU_VAV-FPSR)의 난방계산 알고리즘

(가) 터미널 유닛을 VAV Box, CAV Box w/ Htg. Coil로 선택한 Zone

ZOSAVH(NZ) = ZOSAVMI(NZ)

(나) 터미널 유닛을 VAV Box Fan Powered (Series)로 선택한 Zone

ZOFV(NZ) = ZOSAVMI(NZ)

ZOSAVH(NZ) = ZOFV(NZ)

(다) 터미널 유닛을 VAV Box로 선택한 Zone

ZOQRHH(NZ) = 0

ZOSADBH(NZ) = SADBH

ZOQHC = 0

ZOQPLM = 0

ZOQRHH(NZ) = ZOSAVMI(NZ) x 1.23 x ACF x (ZORMDBH(NZ) - SADBC)

ZOQPLMTRH(NZ) = Σ(RMLIGHPLM x RMMT)/(1.23 x ACF x ZOFV(NZ))

ZOQPLM(NZ) = ZOFV(NZ) x 1.23 x ACF x ZOQPLMTRH(NZ)

(라) 터미널 유닛을 CAV Box w/ Htg. Coil로 선택한 Zone

ZOQPLM(NZ) = 0

ZOSADBH(NZ) = ZORMDBH(NZ) + ZOQH(NZ)/(ZOSAVH(NZ) x 1.23 x ACF)

ZOQHC(NZ) = ZOQRHH(NZ) + ZOQH(NZ) - ZOQPLM(NZ)

전기일 경우 ZOHCKW(NZ) = ZOQHC / 1000

온수1일 경우 ZOHCLPS(NZ)

= ZOQHC(NZ) x 0.86 / (3600 x (HWS1 - HWR1)

온수2일 경우 ZOHCLPS(NZ)

= ZOQHC(NZ) x 0.86 / (3600 x (HWS2 - HWR2)

(마) 각 실별 난방계산(의미없음)

실별난방공급량 RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF

x (ZOSADBH(NZ) - RMDBH(NR))

실별난방과부족 RMHC(NR) = RMQHS(NR) - RMQH(NR)

자. 변풍량 이중덕트 방식(VAV-DD)

(1) 계산조건

(가) 풍량결정기준

냉풍덕트온도(TCD) (℃)

온풍덕트온도(THD) (℃)

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일, · 냉수코일

2. 냉각코일 출구상태(CCSARHMX) 입력

· 냉매코일의 경우 : 기본값 85%

· 냉수코일의 경우 : 기본값 95%

(다) 터미널 유닛의 종류

VAV Box, VAV Box DD Mixing, CAV Box DD Mixing

(2) 냉방계산 알고리즘

(가) ~ (라) : 마. 변풍량 외주부 공기난방 방식, (2) 냉방계산 알고리즘의 (가) ~ (라)와 동일

(마) 냉방풍량 계산

SADBC = TCD로하여 냉방풍량 (SAVC) 계산

(냉풍덕트온도, SADBC)

(바) ~ (서) : 마. 변풍량 외주부 공기난방 방식, (2) 냉방계산

알고리즘의 (바) ~ (서)와 동일

(저) 공조기 냉각코일부하 계산

$$QCC(N) = ACVNEW(N) \times 1.204 \times ACF \times (CCLASAEP - MAEPC(N)) + QSAFB$$

$$ACSAVC_CD = \frac{SYQS(N) + ACVNEW(N) \times 1.23 \times ACF \times (MADBC(N) + SAFTRD + SADTR - SYRMDBC)}{1.23 \times ACF \times (MADBC(N) + SAFTRD - CCLADB)}$$

$$ACSAVC_HD = ACSAVC - ACSAVC_CD$$

(처) AHU SYSTEM 분석치 계산

(3) 터미널 유닛(TU_ VAV-DD)냉방계산 알고리즘

(가) 사용할 수 있는 터미널 유닛의 종류

VAV Box, VAV Box DD Mixing, CAV Box DD Mixing

$$ZOSAVC(NZ) = \sum_{NR(\text{소속zone})} (RMSAVC(NR) \times RMMT(NR))$$

$$ZOQRH(NZ) = 0$$

$$RHKW(NZ) = 0$$

$$RHLPS(NZ) = 0$$

(나) Zone Level에서의 여름철CD, HD 풍량계산

$$ZOSAVC_CD(NZ) = \frac{ZOBKQS(NZ) + ZOSAVC(NZ) \times 1.23 \times ACF \times (MADBC(N) + SAFTRD + SADTR - ZORMDBC(NZ))}{1.23 \times ACF \times (MADBC(N) + SAFTRD - CCLADB)}$$

$$ZOSAVC_HD(NZ) = ZOSAVC(NZ) - ZOSAVC_CD(NZ)$$

(다) 터미널유니트(TU_VAV-DD) 최소풍량(=난방풍량)계산

1. 터미널 유닛을 CAV Box DD Mixing로 선택한 Zone

   $ZOSAVH(NZ) = ZOSAVC(NZ)$

2. 터미널 유닛을 VAV Box, VAV Box DD Mixing로 선택한 Zone

   자료입력화면에서 최소풍량 설정(난방시 고려)값으로부터

   ZOSAVH(NR) =

   - 환기회수 $ZOSAMIXNG(NZ) \times ZOVOL(NZ) / 3.6$
   - 면적당 풍량 $ZOSAMIAR(NZ) \times ZOAR(NZ)$

· % of T.U   ZOSAMI%TU(NZ) x ZOSAVC(NZ) / 100

· 난방취출온도차 ZOHLADT(NZ)로 계산,

ZOQH(NZ) / (1.23 x ACF x ZOHLADT(NZ))

(4) 난방계산 알고리즘

(가) 공조기 난방풍량 계산

ACSAVH = Σ(RMSAVH(NR) x RMMT(NR) x ZOMT(NZ))

(나) 기상자료로부터 OADBH, OAXH를 읽어와서 습공기 상태량 계산

OARHH, OAWBH, OAEPH

(다) 예열코일(외기덕트) 계산

예열코일계산, OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2 구함.

(라) 전열교환기 계산

전열교환기 처리후 공기상태를 계산

OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2

(마) 대표실내온습도 조건(여름)의 상태량 계산

SYRMDBH, SYMRHH   --> SYRMXH, SYRMWBH, SYRMEPH

(바) 급기의 상태량 계산

SADBH = SYRMDBH

SAXH = SysRMXH + SYHUMGS / (0.001204 x ACF x ACSAVH)

SADBC, SAXC --> SAWBC, SARHC, SAEPC (습공기 계산)

(사) 가습기 계산

함수 : F_HUMTR

(아) 냉풍 및 온풍 덕트의 급기풍량 계산

$$\text{ACSAVH_HD} = \frac{\text{SYQH} + \text{ACSAVHx1.23xACFx(SYRMDBH-MADBH2-HUMTR)}}{1.23 \times \text{ACF} \times (\text{THD-MADBH2-HUMTR})}$$

ACSAVH_CD = ACSAVH - ACSAVH_HD

(자) 가열코일용량 계산

QHC = ACSAVH_HD x 1.23 x ACF x (THD - MADBH2 - HUMTR)

전기일 경우 HCKW = QHC / 1000

온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)

증기일 경우 HCKGS = QHC / STQL

• (Warm-up) QHCWARMUP = ACSAVC x 1.23 x ACF x (WARMUPQHTR)

전기일 경우 HCKWWARMUP = QHCWARMUP/1000

온수일 경우 HCLPSWARMUP = QHCWARMUP x 0.86/(3600 x (HWS - HWR)

증기일 경우 HCKGSWARMUP = QHCWARMUP/STQL

(5) 터미널유니트(TU_VAV-DD) 난방계산

(가) 터미널 유닛을 CAV Box DD Mixing로 선택한 Zone

ZOSADBH(NZ) = ZOQH(NZ)/(ZOSAVH(NZ)

x 1.23 x ACF) + ZORMDBH(NZ)

ZOQHC(NZ) = 0

ZO Level에서의 겨울철 CD, HD 풍량계산

$$\text{ZOSAVH_HD(NZ)} = \frac{\text{ZOQH(NZ)} + \text{ZOSAVH(NZ)} \times 1.23 \times \text{ACF} \times (\text{ZORMDBH(NZ)} - \text{MADBH2} - \text{HUMTR})}{1.23 \times \text{ACF} \times (\text{THD} - \text{MADBH2} - \text{HUMTR})}$$

ZOSAVH_CD(NZ)=ZOSAVH(NZ) - ZOSAVH_HD(NZ)

(나) 터미널 유닛을 VAV Box로 선택한 Zone

ZOSADBH(NZ) = MADBH2 + HUMTR

ZOQHC(NZ) = 0

ZONE Level에서의 겨출철 CD, HD 풍량계산

ZOSAVH_HD(NZ) = 0

ZOSAVH_CD(NZ) = ZOSAVH(NZ)

(다) 각 실별 난방계산(의미없음)

1. 각 실별 난방공급열량

RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF

x (ZOSADBH(NZ) - RMDBH(NR))

여기서, RMSAVH = RMSAVC (각 실별풍량)

2. 각 실별 난방 과부족량

RMHC(NR) = RMQHS(NR) - RMQH(NR)

차. 변풍량 외주부 라디에이터 방식(VAV-PH)

(1) 계산조건의 입력

(가) 풍량결정기준

· 냉방취출온도차(CLADT) 입력

· 난방취출온도차(HLADT) 입력

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일, · 냉수코일

2. 냉각코일 출구상태(CCSARHMX) 입력

· 냉매코일의 경우 : 기본값 85%

· 냉수코일의 경우 : 기본값 95%

(다) 터미널 유닛의 종류 :

VAV Box, CAV Box

(2) 냉방계산 알고리즘

(가) ~ (바) : 마. 변풍량 외주부 공기난방 방식(VAV-Rad)의 (2) 냉방계산 알고리즘의 (가) ~ (바)와 동일

(사) 터미널유니트(TU_ VAV-TR)냉방풍량계산

사용할 수 있는 터미널 유닛의 종류 :

VAV Box, CAV Box

$$ZOSAVC(NZ) = \sum_{NR(소속zone)} (RMSAVC(NR) \times RMMT(NR))$$

(아) 공조기냉방풍량 계산

$$ACSAVC = SAVC + \Sigma(ACSAVNEW(NR) \times RMMT(NR) \times ZOMT(NZ))$$

(자) 외기량(OAVC) 계산

(차) 각 시각(N)에 대해 계산하면서 공조기부하(QCC)가 최대로 되는 냉각코일부하 피크시각(CCPKN)을 찾아낸다.

1. VAV Box로 선택된 Zone에 대해서

$$SYVAVQS(N) = \sum (RMQS(NR,N) \times RMMT(NR) \times ZOMT(NZ))$$

$$VAVACVC(N) = SYVAVQS(N) / (1.23 \times ACF \times CLADT)$$

2. CAV Box로 선택된 Zone에 대해서

CAVACVC = Σ (ZOSAVC(NZ) x RMMT(NR) x ZOMT(NZ))

ACVNEW(N) = VAVACVC(N) + CAVACVC

(카) 외기량 재계산

OAVNEW(N) = OAVC x ACVNEW(N)/ACSAVC

(파) ~ (저) : 가. 단일구역 정풍량방식의 (2) 냉방계산 알고리즘의 (자) ~ (버)와 동일

(3) 난방계산 알고리즘

(가) 터미널 유닛(TU_VAV-PAH) 난방풍량 계산

1. 터미널 유닛을 CAV Box을 선택한 Zone

ZOSAVH(NZ) = ZOSAVC(NZ)

2. 터미널 유닛을 VAV Box로 선택한 Zone

자료입력화면에서 최소풍량 설정(난방시 고려)값으로부터

ZOSAVH(NR) =

- 환기회수 ZOSAMIXNG(NZ) x ZOVOL(NZ) / 3.6
- 면적당 풍량 ZOSAMIAR(NZ) x ZOAR(NZ)
- % of T.U ZOSAMI%TU(NZ) x ZOSAVC(NZ) / 100

(나) ACSAVH = Σ (ZOSAVH(NZ) x ZOMT(NZ))

(다) 외기량 계산 (겨울) OAVH 계산

OAVH2 = OAVC x ACSAVH / ACSAVC

if OAVH > OAVH2, then OAVH = OAVH2

주) 겨울외기량은 여름외기량으로부터 풍량비율에 따라 계산됨

(라) 외기 상태량 계산

기상자료로부터 OADBH, OAXH --> OARHH, OAWBH, OAEPH

(마) 대표실내온습도조건의 상태량 계산

SYRMDBH, SYRMRHH를 읽어와서 --> SYRMXH, SYRMWBH, SYRMEPH.

(바) 예열코일(외기덕트) 계산

예열코일계산, OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2 구함

(사) 전열교환기 계산

전열교환기 처리후 공기상태, OADBH2, OAXH2, OARHH2, OAWBH2,

OAEPH2 구함

(아) RA와 OA의 혼합상태 계산

RADBH, OADBH2
RAXH, OAXH2 → MADBH, MAXH → MARHH, MAWBH, MAEPH.

(자) 예열코일(혼합박스) 계산

MADBH2, MAXH2, MARHH2, MAWBH2, MAEPH2

(차) 실내 난방기기가 있슴

난방급기온도(SADBH) = SADBC

SAXH = SYRMXH + SYHUMGS / (0.001204 x ACF x ACSAVH)

(카) 가습기 계산 (HUMTR)

만약 MADBH2 > (SADBH - HUMTR) 이면, SADBH = MADBH2 + HUMTR

QHC = 0, HCKW = 0, HCLPS = 0, HCKGS = 0

또는 가열코일용량 QHC = ACSAVH x 1.23

x ACF x(SADBH - HUMTR - MADBH2)

전기일 경우 HCKW = QHC / 1000

온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)

증기일 경우 HCKGS = QHC / STQL

· (Warm-up) QHCWARMUP = ACSAVC x1.23xACF x (WARMUPQHTR)

전기일 경우 HCKWWARMUP = QHCWARMUP / 1000

온수일 경우 HC_LPSWARMUP = QHCWARMUP x 0.86/(3600x(HWS - HWR)

증기일 경우 HC_KGSWARMUP = QHCWARMUP / STQL

(4) 터미널유니트(TU_VAV-PAH) 난방계산

(가) 계산사항

ZOSAVH(NZ)

ZOQRHH(NZ) = 0

ZOSADBH(NZ)= SADBH

ZOQHC(NZ) = 0

(나) 각 실별 난방계산

실내 난방기기는 없슴

외주부 난방풍량 계산

RMPAH(NR) = RMQH(NR)

RMPAHV(NR) = RMQH(NR) / (1.23 x ACF x HLADT)

PAH = SYQH

PAHV = PAH / (1.23 x ACF x HLADT)

전기일 경우 PAHKW = PAH / 1000

온수(1)일 경우 PAHLPS = PAH x 0.86 / (3600x(HWS1 - HWR1))

온수(2)일 경우 PAHLPS = PAH x 0.86 / (3600x(HWS2 - HWR2))

증기일 경우 PAHKGS = PAH / STQL

카. 패키지형 냉방기 방식(PTAC, PaCkaged Air Conditioner)

(1) 계산조건의 입력

(가) 풍량결정기준

냉방취출온도차(CLADT) 입력

(나) 냉각코일 계산조건

1. 냉각코일의 종류 선정

· 냉매코일

2. 냉각코일 출구상태(CCSARHMX) 입력

· 냉매코일의 경우 : 기본값 85%

(다) 터미널 유닛의 종류 :

· 없음

(2) 냉방계산 알고리즘

(가) ~ (러) : 가. 단일구역 정풍량 방식(CAV-SZ)의 (2) 냉방계산 알고리즘의 (가) ~ (러)와 동일

(머) 공조기 냉각코일부하 계산

QCC(N) = ACSAVC x 1.204 x ACF x (CCSAEP - MAEPC(N))

(버) AHU SYSTEM 분석치 계산

(3) 난방계산 알고리즘

(가) 공조기 급기풍량

ACSAVH = ACSAVC

(나) 외기량 계산

함수 : F_OAVH

(다) 기상자료로부터 OADBH, OAXH를 읽어와서 상태량을 계산 한다.

-> OARHH, OAWBH, OAEPH

(라) 공조기 입력화면으로부터 대표실내온습도조건 SYRMDBH, SYRMRHH로 상태량 계산 -> SYRMXH, SYRMWBH, SYRMEPH

(마) RA와 OA의 혼합상태 계산

RADBH, OADBH2
RAXH, OAXH2
→ MADBH, MAXH → MARHH, MAWBH, MAEPH.

(바) 난방급기온도 계산

1. 실내 난방기기가 있을 때

난방급기온도(SADBH) = SYRMDBH

2. 실내 난방기기가 없을 때

$$\text{난방급기온도(SADBH)} = SYRMDBH + \frac{SYQH}{1.23 \times ACF \times ACSAVH}$$

· 이때 난방급기온도가 55℃이상이면 “Warwing Message”

SAXH = SYRMXH + SYHUMGS / (0.001204 x ACF x ACSAVH)

(사) 가습기 계산

함수 : F_HUMTR

(아) 가열코일 용량 계산

QHC = ACSAVH x 1.23 x ACF x (SADBH - HUMTR - MADBH2)

· 전기일 경우 HCKW = QHC / 1000

· 온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)

· 증기일 경우 HCKGS = QHC / STQL

(자) 각 실별 난방 계산

1. 실내 난방기기가 있을 때

난방기기용량 계산

RMHTG(NR) = RMQH(NR) + RMSAVH(NR) x 1.23

x ACF x (RMDBH(NR) - SADBH)

- 전기일 경우 RMHTGKW(NR) = RMHTG(NR) / 1000
- 온수(1)일 경우 RMHTGLPS(NR) = RMHTG(NR)

  x 0.86/(3600 x (HWS1 - HWR1))
- 온수(2)일 경우 RMHTGLPS(NR) = RMHTG(NR)

  x 0.86/(3600 x (HWS2 - HWR2))
- 증기일 경우 RMHTGKGS(NR) = RMHTG(NR) / STQL

2. 실내 난방기기가 없을 때

각 실별 난방공급열량

RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF x (SADBH - RMDBH(NR))

각 실별 난방과부족량

RMHC(NR) = RMQHS(NR) - RMQH(NR)

여기서, RMSAVH = RMSAVC (각 실별풍량)

(4) FCU System

(가) 팬코일유니트(FCU) 냉방계산

각 실별 냉방계산

실내 냉방기기는 FCU

RMFCUQSC(NR) = RMQPS(NR)

RMFCUQLC(NR) = RMQPLM(NR)

RMFCUQT(NR) = RMQPT(NR)

냉수유량 FCULPSC = RMQPT(NR) x 0.86/(3600 x (CHS - CHR))

(나) 팬코일유니트(FCU) 난방계산

각 실별 난방계산

실내 난방기기는 FCU

RMFCUH(NR) = RMQH(NR)

온수(1)일 경우 FCULPSH1 = RMQH(NR) x 0.86

/ (3600 x (HWS1 - HWR1))

온수(2)일 경우 FCULPSH2 = RMQH(NR) x 0.86

/ (3600 x (HWS2 - HWR2))

파. 난방 전용 방식(HO, Heating Only)

(1) HO 난방계산

(가) 각 실별 난방계산

실내 난방기기는 Room Heater

RMHTER(NR) = RMQH(NR)

전기일 경우 RMHTERKW(NR) = RMHTER(NR) / 1000

온수(1)일 경우

RMHTERLPS(NR) = RMHTER(NR) x 0.86/(3600 x (HWS1 - HWR1))

온수(2)일 경우

RMHTERLPS(NR) = RMHTER(NR) x 0.86/(3600 x (HWS2 - HWR2))

증기일 경우

RMHTERKGS(NR) = RMHTER(NR) / STQL

하. 환기조화기 방식(HVU, Heating & Ventilation Unit)

(1) 계산조건의 입력

(가) 풍량결정기준

냉방취출온도차(CLADT) 입력

(나) 터미널 유닛의 종류 :

없음

(2) 난방계산 알고리즘

(가) HVUV = SYQH / (1.23 x ACF x HLADT)

(나) 외기량(OAVH) 계산

만약 OAVH > HVUV 이면 HVUV = OAVH

HLADT = SYQH / (1.23 x ACF x HVUV)

(다) 기상자료로부터 OADBH, OAXH를 읽어와서 상태량을 계산 한다.

-> OARHH, OAWBH, OAEPH

(라) 공조기 입력화면으로부터 대표실내온습도조건 SYRMDBH, SYRMRHH로 상태량 계산 -> SYRMXH, SYRMWBH, SYRMEPH

(마) 예열코일(외기덕트) 계산

예열코일계산, OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2 구함

(바) 전열교환기 계산

전열교환기 처리후 공기상태

OADBH2, OAXH2, OARHH2, OAWBH2, OAEPH2

(사) RA와 OA의 혼합상태 계산

RADBH, OADBH2
RAXH, OAXH2
→ MADBH, MAXH → MARHH, MAWBH, MAEPH.

(아) 예열코일(혼합박스) 계산

MADBH2, MAXH2, MARHH2, MAWBH2, MAEPH2

(자) 실내 난방기기가 없음

난방급기온도(SADBH) = SYRMDBH + HLADT

· 이때 난방급기온도가 55℃이상이면 "Warwing Message"

SAXH = SYRMXH + SYHUM / (0.001204 x ACF x ACSAVH)

(차) 가습기 계산 (HUMTR)

(카) 가열코일 용량

QHC = HVUV x 1.23 x ACF x (SADBH - HUMTR - MADBH2)

전기일 경우 HCKW = QHC / 1000

온수일 경우 HCLPS = QHC x 0.86/(3600 x (HWS - HWR)

증기일 경우 HCKGS = QHC / STQL

(파) 각 실별 난방계산(의미없음)

각 실별 난방공급열량,

RMQHS(NR) = RMSAVH(NR) x 1.23 x ACF x (SADBH - RMDBH(NR))

각 실별 난방 과부족량

RMHC(NR) = RMQHS(NR) - RMQH(NR)

여기서, RMSAVH = RMSAVC (각 실별풍량)

제 4 장 연구개발목표 달성도 및 대외기여도

제 1 절 연구개발 목표 및 달성도

본 연구의 연구개발 목표 및 달성도는 다음과 같다.

연구개발목표	추진실적	달성도
자료조사 및 분석 • 국내외 S/W개발 현황 조사 • 실무분야의 의견조사 및 분석 • 공조기 제조회사의 자료조사 및 분석	• DOE-2, • HASP, • E-20II • Elite사의 S/W Set • HVAC2 TOOLKIT	100%
GUI 환경 설정 • 프로그램의 사용환경 설정 • 메뉴 구조설정	• 하드웨어 : 586 이상 • 언어 : C/C++ • 주메뉴 8종	100%
기본 TOOL개발 • 습공기 알고리즘 및 S/W개발 • 열전달장치 알고리즘 개발 • 유틸리티 알고리즘개발 • 공조방식 알고리즘 작성	• 습공기선도 (PSYALL) • 과정해석 • 열전달장치(열교환기, 코일) • 유동기기(팬, 펌프) • 응답계수, 전달함수 등 • 16종 작성	100%

제 2 절 연구개발의 대외 기여도

공조시스템의 통합설계 S/W는 설계과정에서의 노력과 시간이 절약될 뿐만아니라 품질과 생산성을 제고시킬 수 있으며 최적설계화에 의한 에너지 절약효과도 매우 크다. 또한 설계과정에서 발생할 수 있는 오차와 오류를 감소시켜 기계설비시설물 등의 부실설계의 가능성을 최소화 할 수 있으며, 설계의 단순 반복과정을 전산화 함으로서 설계의 경제성을 향상시키고 안정성 제고와 미관 및 환경친화적 시설물 확보에 기여할 수 있다.

제 5 장 연구개발결과의 활용계획

본 연구결과는 전체연구기간중 1차년도의 연구결과로 완전한 연구결과가 아님으로 실무적으로 활용하기에는 보다 많은 부분이 보완되어야 할 것이다. 다만, 본 연구결과로 얻어진 습공기선도 해석 및 과정해석 S/W는 건축설비설계회사 및 건설회사와 관련 대학 등에 배포하여 실무 및 교육용으로 활용하도록 하겠으며, 연구결과 작성된 알고리즘은 향후 프로그램으로 개발하여 실무에 활용할 수 있도록 하겠다.

제 6 장 참고문헌

1. J.A. Goff and S. Gratch, 1945, "Thermodynamic properties of moist air", ASHRAE Transaction, Vol. 51.
2. A. Wexler, R.W. Hyland and R.B. Stewart, "Thermodynamic Properties of Dry Air, Moist Air and Water and SI Psychometric Charts", ASHRAE, RP-216 & RP-257.
3. ASHRAE Handbook, 1988, "Fundamentals", American Society of Heating, Refrigerating and Air-Conditioning Engineers.
4. A. Wexler and R.W. Hyland, 1983, "Formulations for the thermodynamic properties of dry air from 173.15 to 473.15 K, and of saturated moist air from 173.15 to 372.15 K, at pressure to 5 MPa", Final Report, ASHRAE Project RP-216, 1980, pt.3.
5. R.M.Nelson and M.Pate, 1986, "A Comparison of Three Moist Air Property Formulations for Computer Applications", ASHRAE Transactions, Vol.92, part1B.
6. 手塚後一, 藤田稔彦, 1983, "濕空氣線圖와 그 應用(1)", 日本 空氣調和・衛生工學 第52卷, 第12號.
7. 手塚後一, 藤田稔彦, 1984, "濕空氣線圖와 그 應用(3)", 日本 空氣調和・衛生工學 第58卷, 第2號.
8. ASHRAE, 1993, "HVAC2 TOOLKIT", ASHRAE.
9. ASHRAE, 1967, "Procedures for simulating the performance of component and system for energy calculation"

여 백

부 록

부록 A. 습공기상태량 계산 프로그램

1. 프로그램

부록 B. 구성기기 계산 프로그램

1. 팬

가. 압력독립형

(1) 프로그램

(2) 계산예제

나. 압력종속형

(1) 프로그램

(2) 계산예제

부록 C. 벽의 열취득 계산 프로그램

1. 프로그램

2. 계산예제

부록 D. 그림자 계산 프로그램

1. 프로그램

2. 계산예제

부록 E. 회귀분석 프로그램

1. 프로그램

2. 계산예제

여 백

부록 A. 습공기상태량 계산 프로그램

1. 프로그램

```
/*****************************************************************************
 *
 *                    PROGRAM MOIST_PROPERTY
 *
 *                                        Filename   : Psyfunc.c
 *                                        Developer  : Kim Doo Chun
 *                                                     Han_il M.E.C
 *                                        CTranslator: Choi Chul Hoon
 *                                        Date:        April ,1998
 *
 *****************************************************************************/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#include "psyhead.h"

/*****************************************************************************
 *                    Function Defintion
 *****************************************************************************/
double presratio_(float *height)
{
    float  ret_val;
    double d__1;
    double c_b2 = 5.256;

    d__1    = (double) ((float)1. - *height * (float)2.2558e-5);
    ret_val = pow(d__1, c_b2);

    return ret_val;
}


double enhance_(float *tdb, float *pr)
{
    static float c1 = (float)1.62867e-5;
    static float c2 = (float)-6.87144e-6;
    static float c3 = (float)6.3217e-7;
    static float c4 = (float)1.11286e-9;
    static float c5 = (float)1.78558e-11;
    static float c6 = (float)-4.71039e-13;
    static float c7 = (float)-7.73271e-15;

    float ret_val, r__1, r__2, r__3, r__4, r__5, r__6, r__7, r__8;

    r__1 = *tdb;
    r__2 = *tdb, r__3 = r__2;
    r__4 = *tdb, r__4 *= r__4;
    r__5 = *tdb, r__6 = r__5, r__5 *= r__5;
    r__7 = *tdb, r__7 *= r__7, r__8 = r__7;
    ret_val = *pr * (float).004 + (float)1. + c1 + c2 * *tdb + c3 * (r__1 *
            r__1) + c4 * (r__3 * (r__2 * r__2)) + c5 * (r__4 * r__4) + c6 * (
            r__6 * (r__5 * r__5)) + c7 * (r__8 * (r__7 * r__7));

    return ret_val;
}

/*
C***************************************************************************
C*    FUNCTION:              DEWPOINT
C*    PURPOSE:               Calculate the dewpoint temperature given
C*                           humidity ratio
C***************************************************************************
C*    INPUT VARIABLES
C*    W            Humidity ratio                                      (-)
```

```
C*
C*    OUTPUT VARIABLES
C*    DewPoint      Dew point temperature of air                      (C)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                              (Pa)
C*************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C*************************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                      (Pa)
C     small         Small number
C*************************************************************************/
double dewpoint_(float *prop, float *w, float *fs)
{
    float small = (float)1e-9;
    float ret_val;
    float pw;

  --prop;

    if(*w < small)
    {
       ret_val = (float)-111.1;
    }
    else
    {
       pw = prop[1] * *w / (*w + (float).62198) / *fs;
       ret_val = sattemp_(&prop[1], &pw);
    }
    return ret_val;
}


/*
C*************************************************************************
C*    FUNCTION:               DRYBULB
C*    PURPOSE:                Calculate the dry bulb temperature of
C*                            moist air from enthalpy and humidity.
C*************************************************************************
C*    INPUT VARIABLES:
C*    H             Enthalpy                                       (J/kg)
C*    W             Humidity ratio                                    (-)
C*
C*    OUTPUT VARIABLES:
C*    Drybulb       Dry bulb temperature                              (C)
C*
C*    PROPERTIES:
C*    CpAir         Specific heat of air                         (J/kg C)
C*    CpVap         Specific heat of water vapor                 (J/kg C)
C*    Hfg           Reference heat of vaporization of water        (J/kg)
C*************************************************************************
C     MAJOR RESTRICTIONS:     Uses perfect gas relationships
C                             Fit for enthalpy of saturated water vapor
C*************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C*************************************************************************/
double drybulb_(float *prop, float *h__, float *w)
{
    float ret_val;

  --prop;
    ret_val = (*h__ - prop[11] * *w) / (prop[2] + prop[4] * *w);
    return ret_val;
}

/*
C*************************************************************************
C*    FUNCTION:               ENTHALPY
C*    PURPOSE:                Calculate the enthalpy of moist air.
C*************************************************************************
```

```
C*    INPUT VARIABLES:
C*    TDB           Dry bulb temperature                        (C)
C*    W             Humidity ratio                              (-)
C*
C*    OUTPUT VARIABLES:
C*    Enthalpy      Enthalpy of moist air                     (J/kg)
C*
C*    PROPERTIES:
C*    CpAir         Specific heat of air                    (J/kg C)
C*    CpVap         Specific heat of water vapor            (J/kg C)
C*    Hfg           Reference heat of vaporization of water   (J/kg)
C***********************************************************************
C     MAJOR RESTRICTIONS      Uses perfect gas relationships
C                             Fit for enthalpy of saturated water vapor
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C***********************************************************************/
double enthalpy_(float *prop, float *tdb, float *w)
{
    float ret_val;
    float hdryair, hsatvap;

  --prop;

    hdryair = prop[2] * *tdb;
    hsatvap = prop[11] + prop[4] * *tdb;
    ret_val = hdryair + *w * hsatvap;
    return ret_val;
}

double enthliq_(float *prop, float *wetbulb)
{
    float ret_val;

  --prop;

    if(*wetbulb > (float)0.)
    {
       ret_val = prop[3] * *wetbulb;
    }
    else
    {
       ret_val = prop[17] * *wetbulb + prop[18];
    }
    return ret_val;
}

/*
C****************************************************************************
C*    FUNCTION:               ENTHSAT
C*    PURPOSE:                Calculate the enthalpy at saturation
C*                            for given dry bulb temperature
C***********************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                        (C)
C*
C*    OUTPUT VARIABLES
C*    EnthSat       Enthalpy at saturation                    (J/kg)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                        (Pa)
C***********************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C***********************************************************************
C     INTERNAL VARIABLES:
C     psat          Saturated water vapor pressure              (Pa)
C     w             Humidity ratio                              (-)
C***********************************************************************/
double enthsat_(float *prop, float *tdb, float *fs)
{
    float ret_val;
    float psat;
```

```
    float w;

  --prop;

    psat    = satpress_(&prop[1], tdb);
    w       = humratio_(&prop[1], &psat, fs);
    ret_val = enthalpy_(&prop[1], tdb, &w);

    return ret_val;
}

/*
C***************************************************************************
C*    FUNCTION:              HUMRATIO
C*    PURPOSE:               Calculate the humidity ratio from water
C*                           vapor pressure and atmospheric pressure
C***************************************************************************
C*    INPUT VARIABLES
C*    Patm         Atmospheric pressure                         (Pa)
C*    Pw           Partial water vapor pressure                 (Pa)
C*
C*    OUTPUT VARIABLES
C*    HumRatio     Humidity ratio                               (-)
C***************************************************************************
C     MAJOR RESRICTIONS:     None
C     REFERENCE:             1989 ASHRAE Handbook - Fundamentals
C***************************************************************************/
double humratio_(float *prop, float *pw, float *fs)
{
    float ret_val;

  --prop;

    ret_val = *pw * (float).62198 * *fs / (prop[1] - *pw * *fs);

    return ret_val;
}

/*
C***************************************************************************
C*    FUNCTION:              HUMTH
C*    PURPOSE:               Calculate the humidity ratio of moist air
C*                           from dry bulb temperature and enthalpy.
C***************************************************************************
C*    INPUT VARIABLES:
C*    H            Enthalpy                                   (J/kg)
C*    TDB          Dry bulb temperature                          (C)
C*
C*    OUTPUT VARIABLES:
C*    HumTH        Humidity ratio                               (-)
C*
C*    PROPERTIES:
C*    CpAir        Specific heat of air                     (J/kg C)
C*    CpVap        Specific heat of water vapor             (J/kg C)
C*    Hfg          Reference heat of vaporization of water    (J/kg)
C***************************************************************************
C     MAJOR RESTRICTIONS:    Uses perfect gas relationships
C                            Fit for enthalpy of saturated water vapor
C     REFERENCE:             1989 ASHRAE Handbook - Fundamentals
C***************************************************************************/
double humth_(float *prop, float *tdb, float *h__)
{
    float ret_val;
  --prop;

    ret_val = (*h__ - prop[2] * *tdb) / (prop[11] + prop[4] * *tdb);

    return ret_val;
}

/*
```

```
C***************************************************************************
C*    FUNCTION:                 RELHUM
C*    PURPOSE:                  Calculate the relative humidity from
C*                              saturation and atmospheric pressures
C***************************************************************************
C*    INPUT VARIABLES
C*    Patm          Atmospheric pressure                           (Pa)
C*    Psat          Saturation pressure                            (Pa)
C*    HumRatio      Humidity ratio                                  (-)
C*
C*    OUTPUT VARIABLES
C*    RelHum        Relative humidity                               (-)
C***************************************************************************
C     REFERENCE:                1989 ASHRAE Handbook - Fundamentals
C***************************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                   (Pa)
C***************************************************************************/
double relhum_(float *prop, float *psat, float *humratio, float *fs)
{
    float ret_val;
    float pw;

  --prop;

    pw = prop[1] * *humratio / (*humratio + (float).62198) / *fs;
    ret_val = pw / *psat;
    if(ret_val > (float)1.)
    {
      ret_val = (float)1.;
    }

    return ret_val;
}


/*
C***************************************************************************
C*    FUNCTION:                 SATPRESS
C*    PURPOSE:                  Calculate saturation pressure of water
C*                              vapor as a function of temperature
C***************************************************************************
C*    INPUT VARIABLES
C*    T             Temperature                                     (C)
C*
C*    OUTPUT VARIABLES
C*    SatPress      Saturation pressure                            (Pa)
C*
C*    PROPERTIES
C*    TKelMult      Multiplying factor to convert user T to Kelvin
C*    TAbsAdd       Additive factor to convert user T to absolute T
C*                  tKel = Prop(TKelMult) * (T + Prop(TAbsAdd))
C*    PaMult        Multiplying factor to convert user P to Pascals
C*    PAbsAdd       Additive factor to convert user P to absolute P
C*                  Pa = Prop(PaMult) * (P + Prop(PAbsAdd))
C***************************************************************************
C     REFERENCE:                1989 ASHRAE Handbook - Fundamentals
C
C                               Hyland, R.W., and A. Wexler.  1983.
C                               Formulations for the thermodynamic
C                               properties of the saturated phases of H2O
C                               from 173.15 K to 473.15 K.  ASHRAE
C                               Transactions, Vol. 89, No. 2A, pp. 500-519
C***************************************************************************
C     INTERNAL VARIABLES:
C     tKel          Temperature in Kelvin                           (K)
C     pascals       Saturation pressure                            (Pa)
C***************************************************************************/
double satpress_(float *prop, float *t)
{
    static float c1 = (float)-5674.5359;
```

```
    static float c2 = (float)6.3925247;
    static float c3 = (float)-.009677843;
    static float c4 = (float)6.2215701e-7;
    static float c5 = (float)2.0747825e-9;
    static float c6 = (float)-9.484024e-13;
    static float c7 = (float)4.1635019;
    static float c8 = (float)-5800.2206;
    static float c9 = (float)1.3914993;
    static float c10 = (float)-.048640239;
    static float c11 = (float)4.1764768e-5;
    static float c12 = (float)-1.4452093e-8;
    static float c13 = (float)6.5459673;

    float ret_val;
    float tkel, pascals;

  --prop;

//C1*** Convert temperature from user units to Kelvin.
    tkel = prop[13] * (*t + prop[14]);
//C1*** If below freezing, calculate saturation pressure over ice.
    if(tkel < (float)273.15)
    {
        pascals = exp(c1 / tkel + c2 + tkel * (c3 + tkel * (c4 + tkel * (c5 + c6 * tkel))) +
                  c7 * log(tkel));
    }
    else
//C1*** If above freezing, calculate saturation pressure over liquid water.
    if(tkel >= (float)273.15)
    {
        pascals = exp(c8 / tkel + c9 + tkel * (c10 + tkel * (c11 + tkel * c12)) + c13 *
                  log(tkel));
    }
    ret_val = pascals / prop[15] - prop[16];

    return ret_val;
}

/*
C***********************************************************************
C*    FUNCTION:                SATTEMP
C*    PURPOSE:                 Calculate the saturation (boiling)
C*                             temperature of water given pressure
C***********************************************************************
C*    INPUT VARIABLES
C*    P             Pressure                                     (Pa)
C*
C*    OUTPUT VARIABLES
C*    SatTemp       Saturation temperature of water vapor         (C)
C***********************************************************************
C     REFERENCE:               1989 ASHRAE Handbook - Fundamentals
C***********************************************************************
C     INTERNAL VARIABLES:
C     tSat          Water temperature guess                       (C)
C     pSat          Pressure corresponding to temp. guess        (Pa)
C     error         Deviation of dependent variable in iteration
C     iter          Iteration counter
C     icvg          Iteration convergence flag
C     F1,F2         Previous values of dependent variable in XITERATE
C     X1,X2         Previous values of independent variable in XITERATE
C***********************************************************************/
double sattemp_(float *prop, float *p)
{
    static int  itmax = 50;
    int i__1;
    float ret_val;

    static int   icvg, iter;
    static float psat, tsat;
    static float f1, f2, error, x1, x2;
```

```
    tsat = (float)100.;
    i__1 = itmax;

    for(iter = 1; iter <= i__1; ++iter)
    {
        psat  = satpress_(prop, &tsat);
        error = *p - psat;
        tsat  = xiterate_(&tsat, &error, &x1, &f1, &x2, &f2, &iter, &icvg)
        if(icvg == 1)
        {
           goto L110;
        }
    }
L110:
    ret_val = tsat;

    return ret_val;
}


/*
C*************************************************************************
C*    FUNCTION:               ENTHSAT
C*    PURPOSE:                Calculate the dry bulb temperature given
C*                            enthalpy at saturation.
C*************************************************************************
C*    INPUT VARIABLES:
C*    HSat          Enthalpy at saturation                        (J/kg)
C*
C*    OUTPUT VARIABLES:
C*    TAirSat       Dry bulb temperature                             (C)
C*************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C*************************************************************************
C     INTERNAL VARIABLES:
C     error         Deviation of dependent variable in iteration
C     iter          Iteration counter
C     icvg          Iteration convergence flag
C     F1,F2         Previous values of dependent variable in XITERATE
C     X1,X2         Previous values of independent variable in XITERATE
C*************************************************************************/
double tairsat_(float *prop, float *hsat, float *fs)
{
    int i__1;
    int itmax = 20;
    int icvg, iter;
    float ret_val;
    float tsat = (float)50.;
    float f1, f2, error, x1, x2;

    if(tsat < (float)-200. || tsat > (float)1e3)
    {
       tsat = (float)50.;
    }
    i__1 = itmax;
    for(iter = 1; iter <= i__1; ++iter)
    {
        error = *hsat - enthsat_(prop, &tsat, fs);
        tsat = xiterate_(&tsat, &error, &x1, &f1, &x2, &f2, &iter, &icvg);
        if(icvg == 1)
        {
           goto L110;
        }
    }
L110:
    ret_val = tsat;

    return ret_val;
}

/*
```

```
C************************************************************************
C*    FUNCTION:               WETBULB
C*    PURPOSE:                Calculate wet bulb temperature from dry
C*                            bulb temperature and humidity ratio
C************************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                          (C)
C*    W             Humidity ratio of air                         (-)
C*
C*    OUTPUT VARIABLES
C*    WetBulb       Wet bulb temperature                          (C)
C*
C*    PROPERTIES:
C*    Patm          Atmospheric pressure                         (Pa)
C*    Hfg           Latent heat of vaporization of water       (J/kg)
C*    CpAir         Specific heat of air                     (J/kg C)
C*    CpVap         Specific heat of water vapor             (J/kg C)
C*    CpWat         Specific heat of water                   (J/kg C)
C************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C************************************************************************
C     INTERNAL VARIABLES:
C     tBoil         Boiling temperature of water at given pressure   (C)
C     psatStar      Saturation pressure at wet bulb temperature      (C)
C     wStar         Humidity  ratio as a function of PsatStar        (-)
C     newW          Humidity ratio calculated with wet bulb guess    (-)
C     error         Deviation of dependent variable in iteration
C     iter          Iteration counter
C     icvg          Iteration convergence flag
C     F1,F2         Previous values of dependent variable in XITERATE
C     X1,X2         Previous values of independent variable in XITERATE
C*************************************************************************/
double wetbulb_(float *prop, float *tdb, float *w, float *fs)
{
    int    itmax = 20;
    int    i__1;
    float  ret_val, r__1;
    int    s_wsfe(), do_fio(), e_wsfe();
    static int    icvg, iter;
    static float  neww;
    static float  psatstar;
    static float  f1, f2, error, wstar, x1, x2, hw, ws;

  --prop;

    r__1 = satpress_(&prop[1], tdb);
    ws   = humratio_(&prop[1], &r__1, fs);

    if((r__1 = ws - *w, fabs(r__1)) <= (float)1e-8)
    {
        ret_val = *tdb;
        return ret_val;
    }
    else
    {
        ret_val = *tdb;
    }
    i__1 = itmax;
    for(iter = 1; iter <= i__1; ++iter)
    {
        if(ret_val < -60) ret_val = -60.;
        if(ret_val >  80) ret_val =  80.;

        psatstar = satpress_(&prop[1], &ret_val);
        wstar    = humratio_(&prop[1], &psatstar, fs);
        hw       = enthliq_(&prop[1], &ret_val);
        neww     = ((prop[11] + prop[4] * ret_val - hw) * wstar - prop[2] * (*tdb
                          - ret_val)) / (prop[11] + prop[4] * *tdb - hw);
        error    = *w - neww;
        ret_val  = xiterate_(&ret_val, &error, &x1, &f1, &x2, &f2, &iter, &icvg);
```

```
      if(icvg == 1)
      {
         goto L900;
      }
   }
L900:
   if(ret_val > *tdb)
   {
      ret_val = *tdb;
   }

   return ret_val;
}


/*
C**************************************************************************
C*    FUNCTION:                 XITERATE
C*    PURPOSE:                  Iterately solves for the value of X which
C*                              satisfies F(X)=0. Given Xi,F(Xi) pairs,
C*                              the subroutine tests for convergence and
C*                              provides a new guess for the value of the
C*                              independent variable X.
C**************************************************************************
C*    INPUT VARIABLES
C*    F0            Current value of the function F(X)
C*    X0            Current value of X
C*    F1,F2         Two previous values of F(Xi)
C*    X1,X2         Two previous values of X
C*
C*        NOTE:     F1,X1,F2,X2 MUST BE STORED AND SAVED IN CALLING
C*                  ROUTINE.  THEY NEED NO INITIALIZATION
C*
C*    ICount        Number of iterations
C*
C*    OUTPUT VARIABLES
C*    XIterate      New estimate of X for F(X)=0
C*    ICvg          Convergence flag  ICvg = 0:  Not converged
C*                                    ICvg = 1:  Converged
C**************************************************************************
C     REFERENCE:                None
C**************************************************************************
C     INTERNAL VARIABLES
C     small         Small number used in place of zero
C     mode          Number of points used in fit
C                   mode = 1:  Use XPerburb to get new X
C                   mode = 2:  Linear equation to get new X
C                   mode > 2:  Quadratic equation to get new X
C     coef(i)       Coefficients for quadratic fit
C                   F(X) = coef(1) + coef(2)*X + coef(3)*X*X
C     check         Term under radical in quadratic solution
C     FiQ,XiQ       Double precision values of Fi,Xi
C     slope         Slope for linear fit
C     tolRel        Relative error tolerance
C     xPerturb      Perturbation applied to X to initialize iteration
C**************************************************************************/
double xiterate_(float *x0, float *f0, float *x1, float *f1, float *x2, float *f2, int
*icount, int *icvg)
{
   static float tolrel = (float)1e-5;
   static float xperturb = (float).1;
   static float small = (float)1e-9;
   float  ret_val, r__1, r__2;
   double d__1;

   static double coef[3];
   static int    mode;
   static double check;
   static float  slope;
   static double f0q, f1q, f2q;
   static float  xother;
```

```
    static double x0q, x1q, x2q;

    r__2 = fabs(*x0);
    if((r__1 = *x0 - *x1, fabs(r__1)) < tolrel * max(r__2,small) && *icount!= 1 || *f0 ==
        (float)0.)
    {
        ret_val = *x0;
        *icvg = 1;
        return ret_val;
    }
    *icvg = 0;
    mode = *icount;
L10:
    if(mode == 1)
    {
        if(fabs(*x0) > small)
        {
            ret_val = *x0 * (xperturb + (float)1.);
        }
        else
        {
            ret_val = xperturb;
        }
    }
    else
    if(mode == 2)
    {
        slope = (*f1 - *f0) / (*x1 - *x0);
        if(slope == (float)0.)
        {
            mode = 1;
            goto L10;
        }
        ret_val = *x0 - *f0 / slope;
    }
    else
    {
        if(*x0 == *x1)
        {
            *x1 = *x2;
            *f1 = *f2;
            mode = 2;
            goto L10;
        }
        else
        if(*x0 == *x2)
        {
            mode = 2;
            goto L10;
        }
        f2q = *f2;
        f1q = *f1;
        f0q = *f0;
        x2q = *x2;
        x1q = *x1;
        x0q = *x0;
        coef[2] = ((f2q - f0q) / (x2q - x0q) - (f1q - f0q) / (x1q - x0q)) / (x2q - x1q);
        coef[1] = (f1q - f0q) / (x1q - x0q) - (x1q + x0q) * coef[2];
        coef[0] = *f0 - (coef[1] + coef[2] * x0q) * x0q;

        if(abs(coef[2]) < 1e-10)
        {
            mode = 2;
            goto L10;
        }
        if((d__1 = (coef[0] + (coef[1] + coef[2] * x1q) * x1q - f1q) / f1q, abs(d__1)) > 1e-4)
        {
            mode = 2;
            goto L10;
        }
```

```
        d__1 = coef[1];
        check = d__1 * d__1 - coef[0] * 4 * coef[2];
        if(check < 0.)
        {
           mode = 2;
           goto L10;
        }
        else
        if(check > 0.)
        {
            ret_val = (-coef[1] + sqrt(check)) / coef[2] / 2;
            xother = -ret_val - coef[1] / coef[2];
            if((r__1 = ret_val - *x0, fabs(r__1)) > (r__2 = xother - *x0, fabs(r__2)))
            {
                ret_val = xother;
            }
       }
       else
       {
           ret_val = -coef[1] / coef[2] / 2;
       }
    }
    if(mode < 3)
    {
       *x2 = *x1;
       *f2 = *f1;
       *x1 = *x0;
       *f1 = *f0;
    }
    else
    {
       if(*f1 * *f0 > (float)0. && *f2 * *f0 > (float)0.)
       {
          if(fabs(*f2) > fabs(*f1))
          {
             *x2 = *x1;
             *f2 = *f1;
          }
       }
       else
       {
          if(*f2 * *f0 > (float)0.)
          {
             *x2 = *x1;
             *f2 = *f1;
          }
       }
       *x1 = *x0;
       *f1 = *f0;
    }
    return ret_val;
}


/*
C************************************************************************
C*    FUNCTION:  SPVOL
C*    PURPOSE:                Calculate dry air specific volume.
C************************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                          (C)
C*    W             Humidity ratio                                (-)
C*
C*    OUTPUT VARIABLES
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                         (Pa)
C*    RAir          Gas constant for air                      (J/kg C)
C*    SpVol         Specific volume of dry air                 (m3/kg)
C*    TAbsAdd       Additive constant to convert user T to absolute T
C************************************************************************
```

```
C     MAJOR RESTRICTIONS:     Perfect gas relationships
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C*************************************************************************
C     INTERNAL VARIABLES:
C     pAir          Partial pressure of dry air                    (Pa)
C*************************************************************************/
double spvol_(float *prop, float *tdb, float *w, float *fs)
{
    float ret_val;
    float wv, pwv;
    float c_b36=(float)1.;

  --prop;

    pwv     = *w * prop[1] / (*w + (float).62198) / *fs;
    wv      = humratio_(&prop[1], &pwv, &c_b36);
    ret_val = prop[12] * (*tdb + prop[14]) * (wv * (float)1.6078 + (float)1.) / prop[1];

    return ret_val;
}

/*******************************************************************************
 *
 *                      PROGRAM MOIST_PROPERTY
 *
 *                              Filename :     Psysbrt.c
 *                              Developer:     Kim Doo Chun
 *                                             Han_il M.E.C
 *                              CTranslator:   Choi Chul Hoon
 *                              Date:           April ,1998
 *
 ******************************************************************************/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#include "psyhead.h"

/*************************************************************************
C*    SUBROUTINE:             TDB_RH
C*    PURPOSE:                Calculate psychrometric properties of
C*                            moist air with input of dry bulb
C*                            temperature and relative humidity.
C*************************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                           (C)
C*    RH            Relative humidity                              (-)
C*
C*    OUTPUT VARIABLES
C*    W             Humidity ratio                                 (-)
C*    H             Enthalpy                                    (J/kg)
C*    TWB           Wet bulb temperature                           (C)
C*    TDP           Dewpoint temperature                           (C)
C*    V             Specific volume of dry air                  (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                     (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                          (Pa)
C*************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C*************************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                  (Pa)
C*************************************************************************/
int tdb_rh(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
    float psat;
    float fs, pr, pw;

  --prop;
```

```
  *errstat = 0;
   pr = presratio_(height);
   prop[1] *= pr;
   fs = enhance_(tdb, &pr);
   psat = satpress_(&prop[1], tdb);
   pw = *rh * psat;
  *w = humratio_(&prop[1], &pw, &fs);
  *h__ = enthalpy_(&prop[1], tdb, w);
  *twb = wetbulb_(&prop[1], tdb, w, &fs);
  *tdp = dewpoint_(&prop[1], w, &fs);

   if(*tdp > *tdb)
   {
      *tdp = *tdb;
   }
   *v = spvol_(&prop[1], tdb, w, &fs);
   prop[1] /= pr;

   return 0;
}

/*************************************************************************
C*    SUBROUTINE:             TDB_TWB
C*    PURPOSE:                Calculate psychrometric properties of
C*                            moist air given dry bulb temperature
C*                            and wet bulb temperature.
C************************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                         (C)
C*    TWB           Wet bulb temperature                         (C)
C*
C*    OUTPUT VARIABLES
C*    W             Humidity ratio                               (-)
C*    RH            Relative humidity                            (-)
C*    H             Enthalpy of air                          (J/kg)
C*    TDP           Dewpoint temperature                         (C)
C*    V             Specific volume of dry air              (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                   (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                        (Pa)
C*    CpAir         Specific heat of air                    (J/kg C)
C*    CpVap         Specific heat of water vapor            (J/kg C)
C*    CpWat         Specific heat of liquid water           (J/kg C)
C*    Hfg           Reference heat of vaporization of water   (J/kg)
C************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C************************************************************************
C     INTERNAL VARIABLES:
C     psat          Saturated water vapor pressure              (Pa)
C     wstar         Saturated humidity ratio at wet bulb temp.   (-)
C************************************************************************/
int tdb_twb(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
   float psat;
   float wstar;
   float fs, hw, pr;

 --prop;

  *errstat = 0;
   pr    = presratio_(height);
   prop[1] *= pr;
   fs    = enhance_(tdb, &pr);
   psat  = satpress_(&prop[1], twb);
   wstar = humratio_(&prop[1], &psat, &fs);
   hw    = enthliq_(&prop[1], twb);
   *w    = ((prop[11] + prop[4] * *twb - hw) * wstar - prop[2] * (*tdb - *twb))
           / (prop[11] + prop[4] * *tdb - hw);
```

```
    if(*w <= (float)0.)
    {
       *w = (float)0.;
    }
    psat = satpress_(&prop[1], tdb);
   *rh   = relhum_(&prop[1], &psat, w, &fs);
   *h__  = enthalpy_(&prop[1], tdb, w);
   *tdp  = dewpoint_(&prop[1], w, &fs);

    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
   *v = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}

/**************************************************************************
C*    SUBROUTINE:               TDB_W
C*    PURPOSE:                  Calculate psychrometric properties of
C*                              moist air given dry bulb temperature
C*                              and humidity ratio.
C*************************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                          (C)
C*    W             Humidity ratio                                (-)
C*
C*    OUTPUT VARIABLES
C*    RH            Relative humidity                             (-)
C*    H             Enthalpy of moist air                      (J/kg)
C*    TWB           Wet bulb temperature                          (C)
C*    TDP           Dewpoint temperature                          (C)
C*    V           Specific volume of dry air                 (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                    (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                         (Pa)
C*************************************************************************
C     REFERENCE:                1989 ASHRAE Handbook - Fundamentals
C************************************************************************/
int tdb_w(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
    float psat;
    float fs, pr;

  --prop;

   *errstat = 0;
    pr = presratio_(height);
    prop[1] *= pr;
    fs = enhance_(tdb, &pr);
    psat = satpress_(&prop[1], tdb);
   *rh = relhum_(&prop[1], &psat, w, &fs);
   *h__ = enthalpy_(&prop[1], tdb, w);
   *twb = wetbulb_(&prop[1], tdb, w, &fs);
   *tdp = dewpoint_(&prop[1], w, &fs);
    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
   *v = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}

/**************************************************************************
C*    SUBROUTINE:               TDB_H
```

```
C*    PURPOSE:                  Calculate psychrometric properties of
C*                              moist air given dry bulb temperature
C*                              and enthalpy.
C***************************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                            (C)
C*    H             Enthalpy of moist air                        (J/kg)
C*
C*    OUTPUT VARIABLES
C*    W             Humidity ratio                                  (-)
C*    RH            Relative humidity                               (-)
C*    TWB           Wet bulb temperature                            (C)
C*    TDP           Dewpoint temperature                            (C)
C*    V             Specific volume of dry air                   (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                      (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                           (Pa)
C*    CpAir         Specific heat of air                        (J/kg C)
C*    CpVap         Specific heat of water vapor                (J/kg C)
C*    Hfg           Reference heat of vaporization of water       (J/kg)
C***************************************************************************
C     REFERENCE:                1989 ASHRAE Handbook - Fundamentals
C***************************************************************************/
int tdb_h(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
    float psat;
    float fs, pr;

  --prop;

   *errstat  = 0;
    pr       = presratio_(height);
    prop[1] *= pr;
    fs       = enhance_(tdb, &pr);
    psat     = satpress_(&prop[1], tdb);
   *w        = (*h__ - prop[2] * *tdb) / (prop[11] + prop[4] * *tdb);
   *rh       = relhum_(&prop[1], &psat, w, &fs);
   *twb      = wetbulb_(&prop[1], tdb, w, &fs);
   *tdp      = dewpoint_(&prop[1], w, &fs);

    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
   *v        = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}

/*
C***************************************************************************
C*    SUBROUTINE:               TDB_TDP
C*    PURPOSE:                  Calculate psychrometric properties of
C*                              moist air with input of dry bulb
C*                              temperature and dew point temperature.
C***************************************************************************
C*    INPUT VARIABLES
C*    TDB           Dry bulb temperature                            (C)
C*    TDP           Dewpoint temperature                            (C)
C*
C*    OUTPUT VARIABLES
C*    W             Humidity ratio                                  (-)
C*    H             Enthalpy                                     (J/kg)
C*    TWB           Wet bulb temperature                            (C)
C*    RH            Relative humidity                               (-)
C*    V           Specific volume of dry air                     (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                      (-)
C*
```

```
C*    PROPERTIES
C*    Patm          Atmospheric pressure                          (Pa)
C***************************************************************************
C     REFERENCE:                1989 ASHRAE Handbook - Fundamentals
C***************************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                  (Pa)
C**************************************************************************/
int tdb_tdp(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
    float psat;
    float fs, pr, pw;

  --prop;

    *errstat = 0;
    pr       = presratio_(height);
    prop[1] *= pr;
    fs       = enhance_(tdb, &pr);
    psat     = satpress_(&prop[1], tdb);
    if(*tdp <= (float)-110.)
    {
       pw = (float)0.;
       *w  = (float)0.;
    }
    else
    {
       pw = satpress_(&prop[1], tdp);
       *w  = humratio_(&prop[1], &pw, &fs);
    }
    *rh  = pw / psat;
    *h__ = enthalpy_(&prop[1], tdb, w);
    *twb = wetbulb_(&prop[1], tdb, w, &fs);
    *v   = spvol_(&prop[1], tdb, w, &fs);

    prop[1] /= pr;

    return 0;
}

/***************************************************************************
C*    SUBROUTINE:               W_RH
C*    PURPOSE:                  Calculate psychrometric properties of
C*                              moist air with input of dry bulb
C*                              temperature and relative humidity.
C***************************************************************************
C*    INPUT VARIABLES
C*    W             Humidity ratio                                 (_)
C*    RH            Relative humidity                              (-)
C*
C*    OUTPUT VARIABLES
C*    TDB           Dry bulb temperaure                            (C)
C*    H             Enthalpy                                    (J/kg)
C*    TWB           Wet bulb temperature                           (C)
C*    TDP           Dewpoint temperature                           (C)
C*    V             Specific volume of dry air                  (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                     (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                          (Pa)
C***************************************************************************
C     REFERENCE:                1989 ASHRAE Handbook - Fundamentals
C***************************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                  (Pa)
C**************************************************************************/
int rh_w(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat,float *height)
{
    float small = (float)1e-6;
```

```
    float tdb_temp__;
    float fs, pr, pw, fs_new__;
    float pws;
    float c_b40 =(float)25.;

  --prop;

   *errstat = 0;
    if(*rh <= (float)0. && *w <= (float)0.)
    {
       goto L808;
    }
    if(*rh <= (float)0. || *w <= (float)0.)
    {
       goto L888;
    }
    if(*rh < small)
    {
      *rh = small;
    }
    pr   = presratio_(height);
    prop[1] *= pr;
    fs   = enhance_(&c_b40, &pr);
    pw   = *w * prop[1] / ((*w + (float).62198) * fs);
    pws  = pw / *rh;
    tdb_temp__ = sattemp_(&prop[1], &pws);
    fs_new__ = enhance_(&tdb_temp__, &pr);
    pw   = *w * prop[1] / ((*w + (float).62198) * fs_new__);
    pws  = pw / *rh;
    *tdb = sattemp_(&prop[1], &pws);
    fs   = enhance_(tdb, &pr);
    *h__ = enthalpy_(&prop[1], tdb, w);
    *twb = wetbulb_(&prop[1], tdb, w, &fs);
    *tdp = dewpoint_(&prop[1], w, &fs);

    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
   *v = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
L808:
    return 0;
L888:
    return 0;
}

/************************************************************************
C*    SUBROUTINE:               W_H
C*    PURPOSE:                  Calculate psychrometric properties of
C*                              moist air with input of Enthalpy and
C*                              Humidity Ratio.
C***********************************************************************
C*    INPUT VARIABLES
C*    H             Enthalpy                                      (J/kg)
C*    W             Humidity ratio                                   (-)
C*
C*    OUTPUT VARIABLES
C*    TDB           Dry bulb temperature                             (C)
C*    TWB           Wet bulb temperature                             (C)
C*    TDP           Dewpoint temperature                             (C)
C*    RH            Relative humidity                                (-)
C*    V             Specific volume of dry air                   (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                       (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                            (Pa)
C***********************************************************************
C     REFERENCE:                1989 ASHRAE Handbook - Fundamentals
```

```
C****************************************************************************
C     INTERNAL VARIABLES:
C     pw              Partial water vapor pressure                     (Pa)
C****************************************************************************
*/
int w_h(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
    float psat;
    float fs, pr;

  --prop;

    *errstat = 0;
    pr   = presratio_(height);
    prop[1] *= pr;
    *tdb = drybulb_(&prop[1], h__, w);
    fs   = enhance_(tdb, &pr);
    psat = satpress_(&prop[1], tdb);
    *rh  = relhum_(&prop[1], &psat, w, &fs);
    *twb = wetbulb_(&prop[1], tdb, w, &fs);
    *tdp = dewpoint_(&prop[1], w, &fs);

    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
   *v        = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}

/*
C****************************************************************************
C*    SUBROUTINE:             W_WTB
C*    PURPOSE:                Calculate psychrometric properties of
C*                            moist air with input of Humidity Ratio
C*                             & Wet bulb Temperature
C****************************************************************************
C*    INPUT VARIABLES
C*    W               Humidity ratio                                   (-)
C*    TWB             Wet bulb temperature                             (C)
C*
C*    OUTPUT VARIABLES
C*    TDB             Dry bulb temperature                             (C)
C*    TDP             Dewpoint temperature                             (C)
C*    H               Enthalpy                                      (J/kg)
C*    RH              Relative humidity                                (-)
C*    V               Specific volume of dry air                   (m3/kg)
C*    ErrStat         Error flag (0=ok, 1=error)                       (-)
C*
C*    PROPERTIES
C*    Patm            Atmospheric pressure                            (Pa)
C****************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C****************************************************************************
C     INTERNAL VARIABLES:
C     pw              Partial water vapor pressure                    (Pa)
C***************************************************************************/
int w_twb(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
    float small = (float)1e-6;
    float r__1;
    float psat;
    float psatstar;
    float error, wstar;
    float fs, hw, pr, fs_new__;

  --prop;
```

```
   *errstat = 0;
    pr    = presratio_(height);
    prop[1] *= pr;
    hw    = enthliq_(&prop[1], twb);
    psatstar = satpress_(&prop[1], twb);
    fs    = enhance_(twb, &pr);
L1:
    wstar = humratio_(&prop[1], &psatstar, &fs);
   *tdb   = (prop[2] * *twb + (prop[4] * *twb + prop[11] - hw) * wstar - (prop[11] - hw) * *w)
            / (prop[2] + prop[4] * *w);
    fs_new__ = enhance_(tdb, &pr);
    error = (r__1 = fs_new__ - fs, fabs(r__1));
    fs    = fs_new__;

    if(error > small)
    {
       goto L1;
    }

    psat = satpress_(&prop[1], tdb);
   *rh   = relhum_(&prop[1], &psat, w, &fs);
   *tdp  = dewpoint_(&prop[1], w, &fs);
    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
    *h__ = enthalpy_(&prop[1], tdb, w);
    *v   = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}

/*
C****************************************************************************
C*    SUBROUTINE:              RH_H
C*    PURPOSE:                 Calculate psychrometric properties of
C*                             moist air with input of Relative Humidity
C*                              & Specific Enthalpy
C****************************************************************************
C*    INPUT VARIABLES
C*    RH            Relative humidity                              (-)
C*    H             Enthalpy                                    (J/kg)
C*
C*    OUTPUT VARIABLES
C*    TDB           Dry bulb temperature                           (C)
C*    TWB           Wet bulb temperature                           (C)
C*    TDP           Dewpoint temperature                           (C)
C*    W             Humidity ratio                                 (-)
C*    V             Specific volume of dry air                  (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                     (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                          (Pa)
C****************************************************************************
C     REFERENCE:               1989 ASHRAE Handbook - Fundamentals
C****************************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                  (Pa)
C****************************************************************************/
int rh_h(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat, float *height)
{
    int   i__1;
    int   itmax = 20;
    int   icvg;
    int   iter;
    float r__1;
    float newh;
    float f1, f2, error;
```

```
    float x1, x2;
    float fs, pr, pw;

  --prop;

   *errstat = 0;
    pr   = presratio_(height);
    prop[1] *= pr;
    r__1 = *h__ / prop[2];
   *tdb  = min(r__1,(float)30.);
    i__1 = itmax;

    for(iter = 1; iter <= i__1; ++iter)
    {
        pw     = satpress_(&prop[1], tdb) * *rh;
        fs     = enhance_(tdb, &pr);
        *w     = humratio_(&prop[1], &pw, &fs);
        newh   = enthalpy_(&prop[1], tdb, w);
        error  = *h__ - newh;
        *tdb   = xiterate_(tdb, &error, &x1, &f1, &x2, &f2, &iter, &icvg);
        if(icvg == 1)
        {
            goto L900;
        }
    }
L900:
   *twb = wetbulb_(&prop[1], tdb, w, &fs);
   *tdp = dewpoint_(&prop[1], w, &fs);
    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
   *v        = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}

/*
C***************************************************************************
C*    SUBROUTINE:             RH_TWB
C*    PURPOSE:                Calculate psychrometric properties of
C*                            moist air with input of Relative Humidity
C*                             & Wet bulb temperature
C***************************************************************************
C*    INPUT VARIABLES
C*    RH            Relative humidity                             (-)
C*    TWB           Wet bulb temperature                          (C)
C*
C*    OUTPUT VARIABLES
C*    TDB           Dry bulb temperature                          (C)
C*    TDP           Dewpoint temperature                          (C)
C*    W             Humidity ratio                                (-)
C*    H             Enthalpy                                   (J/kg)
C*    V             Specific volume of dry air                (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                    (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                         (Pa)
C***************************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C***************************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                 (Pa)
C***************************************************************************/
int rh_twb(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp,
float *v, int *errstat,float *height)
{
    int   itmax = 20;
    int   i__1;
    int   icvg, iter;
```

```
    float f1, f2, error, wstar;
    float x1, x2;
    float fs, hw, pr, pw, newtdb, pwstar;

  --prop;

   *errstat = 0;
    pr   = presratio_(height);
    prop[1] *= pr;
    hw   = enthliq_(&prop[1], twb);
    pwstar = satpress_(&prop[1], twb);
    *tdb = *twb;
    i__1 = itmax;

    for(iter = 1; iter <= i__1; ++iter)
    {
        fs       = enhance_(tdb, &pr);
        wstar    = humratio_(&prop[1], &pwstar, &fs);
        pw       = satpress_(&prop[1], tdb) * *rh;
        *w       = humratio_(&prop[1], &pw, &fs);
        newtdb   = (prop[2] * *twb + (prop[4] * *twb + prop[11] - hw) * wstar -
                   (prop[11] - hw) * *w) / (prop[2] + prop[4] * *w);
        error    = newtdb - *tdb;
        *tdb     = xiterate_(&newtdb, &error, &x1, &f1, &x2, &f2, &iter, &icvg)

        if(icvg == 1)
        {
          goto L900;
        }
    }
L900:
   *tdp = dewpoint_(&prop[1], w, &fs);
    if(*tdp > *tdb)
    {
       *tdp = *tdb;
    }
   *h__ = enthalpy_(&prop[1], tdb, w);
   *v   = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}

/*
C***********************************************************************
C*    SUBROUTINE:             H_TWB
C*    PURPOSE:                Calculate psychrometric properties of
C*                            moist air with input of Enthalpy
C*                             & Wet bulb temperature
C***********************************************************************
C*    INPUT VARIABLES
C*    H             Enthalpy                                      (J/kg)
C*    TWB           Wet bulb temperature                             (C)
C*
C*    OUTPUT VARIABLES
C*    TDB           Dry bulb temperature                             (C)
C*    TDP           Dewpoint temperature                             (C)
C*    RH            Relative humidity                                (-)
C*    W             Humidity ratio                                   (-)
C*    V             Specific volume of dry air                     (m3/kg)
C*    ErrStat       Error flag (0=ok, 1=error)                       (-)
C*
C*    PROPERTIES
C*    Patm          Atmospheric pressure                            (Pa)
C***********************************************************************
C     REFERENCE:              1989 ASHRAE Handbook - Fundamentals
C***********************************************************************
C     INTERNAL VARIABLES:
C     pw            Partial water vapor pressure                    (Pa)
C***********************************************************************
*/
```

```
int h_twb(float *prop, float *tdb, float *w, float *rh, float *h__, float *twb, float *tdp
float *v, int *errstat,float *height)
{
    int   itmax = 20;
    int   i__1;
    int   icvg, iter;
    float psat;
    float f1, f2, error, wstar;
    float x1, x2;
    float fs, hw, pr, newtdb;
    float pwstar;

  --prop;

   *errstat = 0;
    pr       = presratio_(height);
    prop[1]*= pr;
    hw       = enthliq_(&prop[1], twb);
    pwstar   = satpress_(&prop[1], twb);
    *tdb     = *twb;
    i__1     = itmax;

    for(iter = 1; iter <= i__1; ++iter)
    {
        fs    = enhance_(tdb, &pr);
        wstar = humratio_(&prop[1], &pwstar, &fs);
       *w     = (*h__ - prop[2] * *tdb) / (prop[11] + prop[4] * *tdb);
        if(*w < (float)0.)
        {
           *w = (float)0.;
        }
        newtdb = (prop[2] * *twb + (prop[4] * *twb + prop[11] - hw) * wstar -
                           (prop[11] - hw) * *w) / (prop[2] + prop[4] * *w);
        error = newtdb - *tdb;
       *tdb = xiterate_(&newtdb, &error, &x1, &f1, &x2, &f2, &iter, &icvg);
        if(icvg == 1)
        {
           goto L900;
        }
    }
L900:
   *tdp  = dewpoint_(&prop[1], w, &fs);
    psat = satpress_(&prop[1], tdb);
   *rh   = relhum_(&prop[1], &psat, w, &fs);
   *h__  = enthalpy_(&prop[1], tdb, w);
   *v    = spvol_(&prop[1], tdb, w, &fs);
    prop[1] /= pr;

    return 0;
}


/*********************************************************************************
 *
 *                           PROGRAM MOIST_PROPERTY
 *                               Test Program
 *
 *                                               Filename :     psytest.c
 *                                               Developer:     Kim Doo Chun
 *                                                              Han_il M.E.C
 *                                               CTranslator:   Choi Chul Hoon
 *                                               Date:          April ,1998
 *********************************************************************************/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <conio.h>

#include "psyhead.h"
```

```
#define MsgTdb "Enter Dry bulb Temperature : Tdb (C)"
#define MsgRh  "Enter Relative Humidity : Rh (%)"
#define MsgTwb "Enter Wet bulb Temperature : Twb (C)"
#define MsgW   "Enter Humidity Ratio : W (g/kg)"
#define MsgH   "Enter Specific Enthalpy : h (kJ/kg)"
#define MsgTdp "Enter Dew point Temperature : Tdp (C)"

#define MsgAtm "Enter the Elevation of City (m) : for Standard Atmospheric Pressure - 0.0"
#define MsgMN  "Enter Menu Number:    (1)Tdb_Rh (2)Tdb_Twb (3)Tdb_W   (4)Tdb_H (5)Tdb_Tdp\n
               (6)Rh_W   (7)W_H     (8)W_Twb  (9)Rh_H  (10)Rh_Twb "

#define MsgContinue "Continue or Quit: \n(1) Continue(same height) (2) Continue(different
height) (3) quit"
#define MsgVal "%10.3f =< %s =< %10.3f"

float prop[19] =
{
      101325.,1006.,4186.,1805.,4186.,1.82e-5,0.00144,0.026,
      0.604 , 998.,2.501e6,287.055,1.,273.15,1.,0.,2093.,-333430.
};

void main(void)
{
     int    ch;
     int    mn;            // Menu No
     int    errstat;
     int    smenu;
     char   str[100]="";
     char   temp[100]="";
     float  min1,max1,min2,max2;
     float  tdb, w, rh, h, twb, tdp, v;
     float  tdb2, w2, rh2, h2, twb2, tdp2, v2;

     float height;
     float wmax,wmin;
     float hmax,hmin;
     float rhmin,rhmax;
     float twbmin,twbmax;

     float tdb_w0;
     float twb_w0;
     float h_w0;

     clrscr();
     printf("\n\n%s\n",MsgAtm);
     height = atof(gets(temp));
     printf("height = %10.2f\n",height);
     for(;;)
     {
        printf("%s\n",MsgMN);
        gets(str);
        mn = atoi(str);

        switch(mn)
        {
 // tdh_rh
           case  1:
                 min1 = -40.00;
                 max1 =  80.00;
                 min2 =   0.00;
                 max2 = 100.00;

                 printf("%s %10.3f =< %s =< %10.3f\n",MsgTdb,min1,"tdb",max1);
                 gets(str);
                 tdb = atof(str);

                 printf("%s %10.3f =< %s =< %10.3f\n",MsgRh,min2,"rh",max2);
                 gets(str);
                 rh = atof(str);
                 rh = rh * 0.01;
```

```
		tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
		break;

// tdb_twb
	case	2:
		min1 = -40.00;
		max1 = 80.00;
		printf("%s %10.3f =< %s =< %10.3f\n",MsgTdb,min1,"tdb",max1);
		gets(str);
		tdb  = atof(str);
		w    = 0.000;

		tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
		min2 = twb+0.001;
		max2 = tdb;

		printf("%s %10.3f =< %s =< %10.3f\n",MsgTwb,min2,"twb",max2);
		gets(str);
		twb  = atof(str);

		tdb_twb(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height)
		break;

// tdb_w
	case	3:
		min1 = -40.00;
		max1 = 80.00;
		printf("%s %10.3f =< %s =< %10.3f\n",MsgTdb,min1,"tdb",max1);
		gets(str);
		tdb  = atof(str);

		rh   = 1.00;
		tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
		min2 = 0.00;
		max2 = w;
		max2 = w * 1000.-0.0005;

		printf("%s %10.3f =< %s =< %10.3f\n",MsgW,min2,"w",max2);
		gets(str);
		w  = atof(str)*0.001;

		tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
		break;

// tdb_h
	case	4:
		min1 = -40.00;
		max1 = 80.00;
		printf("%s %10.3f =< %s =< %10.3f\n",MsgTdb,min1,"tdb",max1);
		gets(str);
		tdb  = atof(str);
		rh   = 1.0;

		tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
		min2 = (prop[1]*tdb)*0.001;
		max2 = h * 0.001;

		printf("%s %10.3f =< %s =< %10.3f\n",MsgH,min2,"h",max2);
		gets(str);
		h  = atof(str)*1000;
		tdb_h(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
		break;

// tdb_tdp
	case	5:
		min1 = -40.00;
		max1 = 80.00;
		printf("%s %10.3f =< %s =< %10.3f\n",MsgTdb,min1,"tdb",max1);
		gets(str);
		tdb  = atof(str);
```

```
            min2 = -110.00;
            max2 = tdb;

            printf("%s %10.3f =< %s =< %10.3f\n",MsgTdp,min2,"tdp",max2);
            gets(str);
            tdp = atof(str);
            tdb_tdp(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
            break;

// rh_w
        case 6:
            tdb   = 80;
            rh    = 1.0;
            tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
            wmin  = 0.0;
            wmax  = w*1000;
            rhmin = 0.001;
            rhmax = 100.0;

            printf("First Parameter:  (1) RH  (2) W\n");
            smenu = atoi(gets(str));

            switch(smenu)
            {
                // (rh->w)
                case 1:
                     printf("%s %10.3f =< %s =< %10.3f\n",MsgRh,rhmin,"rh",rhmax);
                     gets(str);
                     rh2  = atof(str)*0.01;

                     tdb =-40;
                     tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height)
                     wmin = w*1000;

                     rh  = atof(str)*0.01;
                     tdb = 80;
                     tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height)
                     wmax = w*1000;

                     printf("%s %10.3f =< %s =< %10.3f\n",MsgW,wmin,"w",wmax);
                     gets(str);
                     w  = atof(str)*0.001;
                     rh = rh2;

                     rh_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                     break;

                // (w->rh)
                case 2:
                     printf("%s %10.3f =< %s =< %10.3f\n",MsgW,wmin,"w",wmax);
                     gets(str);
                     w2  = atof(str)*0.001;

                     w    = w2;
                     tdb  = 80;
                     tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                     rhmin = rh*100.;

                     if(abs(100.-rhmin)<=1.E-3)
                     {
                        printf("Singular point - only one RH \n",rhmax,"rh",rhmin);
                     }
                     printf("%s %10.3f =< %s =< %10.3f\n",MsgRh,rhmin,"rh",rhmax);
                     rh = atof(gets(str))*0.01;
                     w  = w2;

                     rh_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                     break;
            }
            break;
```

```
// w_h
        case  7:
              tdb =  60.0;
              rh  =   1.0;
              tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
              wmax = w*1000;
              hmax = h*0.001;

              tdb = -40.0;
              rh  =   0.00001;
              tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
              wmin = w*1000;
              hmin = h*0.001;

              printf("First Parameter:  (1) W  (2) H\n");
              smenu = atoi(gets(str));
              switch(smenu)
              {
                  case 1:
                       printf("%s %10.3f =< %s =< %10.3f\n",MsgW,wmin,"w",wmax);
                       w = atof(gets(str))*0.001;

                       tdb = 80;
                       tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                       hmax = h * 0.001;

                       rh  = 1.0 ;
                       rh_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                       hmin = h * 0.001;

                       if(abs(hmax-hmin)<=1E-4)
                       {
                          printf("Singular point - only one H");
                       }
                       printf("%s %10.3f =< %s =< %10.3f\n",MsgH,hmin,"h",hmax);
                       h = atof(gets(str))*1000;

                       w_h(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                       break;

                  case 2:
                       printf("%s %10.3f =< %s =< %10.3f\n",MsgH,hmin,"h",hmax);
                       h = atof(gets(str))*1000;
                       h2=h;

                       rh       = 1.0;
                       tdb_w0   = prop[1]*60.;
                       rh_h(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                       wmax = w*1000.;

                       if(h < tdb_w0)      wmin = 1.e-9;
                       else
                       {
                          tdb = 60.;
                          tdb_h(prop,  &tdb,  &w,  &rh,  &h,  &twb,  &tdp,  &v,  &errstat,
                                &height);
                          wmin = w*1000.;
                       }
                       if(abs(wmax-wmin)<=1.E-4)
                       {
                          printf("Singular point - only one H\n");
                       }
                       printf("%s %10.3f =< %s =< %10.3f\n",MsgW,wmin,"w",wmax);
                       w = atof(gets(str))*0.001;
                       h = h2;
                       w_h(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                       break;
              }
              break;
```

```
// w_twb
          case  8:
                tdb =  60.0;
                rh  =   1.0;
                tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                wmax = w*1000;

                tdb = -40.0;
                rh  =   0.00001;
                tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                wmin   = w*1000;
                twbmin = twb;
                twbmax = 60.;

                printf("First Parameter:  (1) W  (2) TWB\n");
                smenu = atoi(gets(str));
                switch(smenu)
                {
                    case 1:     // w->twb
                         printf("%s %10.3f =< %s =< %10.3f\n",MsgW,wmin,"w",wmax);
                         w   = atof(gets(str))*0.001;

                         tdb=60;
                         tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                         twbmax = twb;

                         rh=1.0;
                         rh_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                         twbmin = twb;

                         if(abs(twbmax-twbmin)<=1e-3)
                         {
                            printf("Singular point - only one Twb\n");
                         }

                         printf("%s %10.3f =< %s =< %10.3f\n",MsgTwb,twbmin,"Twb",twbmax);
                         twb = atof(gets(str));

                         w_twb(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                         break;

                    case 2:
                         printf("%s %10.3f =< %s =< %10.3f\n",MsgTwb,twbmin,"Twb",twbmax);
                         twb  = atof(gets(str));
                         twb2 = twb;

                         rh=1.0;
                         rh_twb(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height)
                         wmax = w*1000.;

                         tdb = 60;
                         w   = 0.0;
                         tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                         twb_w0 = twb;

                         if(twb <= twb_w0)
                         {
                            wmin = 1.e-9;
//                          printf("twb, twb_w0, twb_low = %10.3f, %10.3f,
                                    %10.3f\n",twb,twb_w0,wmin);
                         }
                         if(wmax-wmin <= 1e-4)
                         {
                            printf("Singular point - only one W\n");
                         }

                         printf("%s %10.3f =< %s =< %10.3f\n",MsgW,wmin,"w",wmax);
                         w   = atof(gets(str))*0.001;
                         twb = twb2;

                         w_twb(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
```

```
                        break;
                }
                break;

// rh_h
        case  9:
                rhmin = 0.01;
                rhmax = 100.;
                tdb   = 60.;
                rh    = 1.0;
                tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                hmax = h * 0.001;
                hmin = -40 * prop[1]*0.001;

                printf("First Parameter:  (1) RH (2) H\n");
                smenu = atoi(gets(str));
                switch(smenu)
                {
                    case 1:      // rh->h
                        printf("%s %10.3f =< %s =< %10.3f\n",MsgRh,rhmin,"rh",rhmax);
                        rh   = atof(gets(str))*0.01;
                        rh2  = rh;

                        tdb =  60.0;
                        rh  =  rh;
                        tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                        hmax = h*0.001;

                        tdb = -40.0;
                        rh  =  rh;
                        tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                        hmin = h*0.001;

                        printf("%s %10.3f =< %s =< %10.3f\n",MsgH,hmin,"h",hmax);
                        h   = atof(gets(str))*1000;
                        rh  = rh2;
                        break;

                    case 2: // h->rh
                        printf("%s %10.3f =< %s =< %10.3f\n",MsgH,hmin,"h",hmax);
                        h   = atof(gets(str))*1000;

                        if((h-hmax)<=1e-2)
                        {
                           printf("Singular point - only one H\n");
                        }
                        rhmax = 100.;
                        h_w0  = 60.0 * prop[1];

                        if(h <= h_w0)
                        {
                           rhmin = 0.001;
                        }
                        else
                        {
                           tdb = 60.0;
                           tdb_h(prop,  &tdb,  &w,  &rh,  &h,  &twb,  &tdp,  &v,  &errstat,
                                 &height);
                           rhmin = rh * 100.;
                        }

                        printf("%s %10.3f =< %s =< %10.3f\n",MsgRh,rhmin,"rh",rhmax);
                        rh   = atof(gets(str))*0.01;
                        break;
                }
                rh_h(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                break;

 // rh_twb
        case 10:
                rhmin = 0.001;
```

```
        rhmax = 100.0;
        tdb   = -40.;
        rh    = 0.00001;
        tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
        twbmin = twb;
        twbmax = 60;

        printf("First Parameter:  (1) RH  (2) TWB\n");
        smenu = atoi(gets(str));
        switch(smenu)
        {
            case 1:
                printf("%s %10.3f =< %s =< %10.3f\n",MsgRh,rhmin,"rh",rhmax);
                rh   = atof(gets(str))*0.01;

                tdb = 60;
                tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                twbmax = twb;

                tdb = -40;
                tdb_rh(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                twbmin = twb;

                printf("%s %10.3f =< %s =< %10.3f\n",MsgTwb,twbmin,"twb",twbmax);
                twb   = atof(gets(str));
                break;

            case 2:
                printf("%s %10.3f =< %s =< %10.3f\n",MsgTwb,twbmin,"twb",twbmax);
                twb   = atof(gets(str));
                twb2 = twb;

                tdb = 60;
                w   = 0.0;
                tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
                twb_w0=twb;

                if(twb <= twb_w0)
                {
                   rhmin = 0.001;
                }
                else
                {
                   tdb_twb(prop,  &tdb,  &w,  &rh,  &h,  &twb,  &tdp,  &v,  &errstat,
                           &height);
                   rhmin = rh * 100.;
                }

                rhmax = 100.;
                if(abs(rhmax-rhmin)<=1e-3)
                {
                   printf("Singular point - only one RH\n");
                   rh = rh * 0.01;
                }

                printf("%s %10.3f =< %s =< %10.3f\n",MsgRh,rhmin,"rh",rhmax);
                rh   = atof(gets(str))*0.01;
                twb  = twb2;

                break;
        }
        rh_twb(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
        break;
}
rh = rh * 100.;
h  = h  * 0.001;
w  = w  * 1000.;
printf("%8s %8s %8s %8s %8s %8s %8s\n","tdb","w", "rh", "h", "twb", "tdp", "v");
printf("%8.2f %8.3f %8.2f %8.2f %8.2f %8.2f %8.3f\n",tdb, w, rh, h, twb, tdp, v);

printf("%s\n",MsgContinue);
```

```
        ch = getche();

        if((char)ch == '3') break;
        else
        if((char)ch == '2')
        {
           printf("\n\n%s\n",MsgAtm);
           height = atof(gets(temp));
           printf("height = %10.2f\n",height);
        };
     }
}



/*******************************************************************************
 *
 *                    PROGRAM MOIST_PROPERTY
 *
 *                                          Filename :    psyhead.h
 *                                          Developer:    Kim Doo Chun
 *                                                        Han_il M.E.C
 *                                          Date:         April ,1998
 *
 *******************************************************************************/
double presratio_  (float *height);
double enhance_    (float *tdb, float *pr);
double dewpoint_   (float *prop, float *w, float *fs);
double drybulb_    (float *prop, float *h__, float *w);
double enthalpy_   (float *prop, float *tdb, float *w);
double enthliq_    (float *prop, float *wetbulb);
double enthsat_    (float *prop, float *tdb, float *fs);
double humratio_   (float *prop, float *pw, float *fs);
double humth_      (float *prop, float *tdb, float *h__);
double relhum_     (float *prop, float *psat, float *humratio, float *fs);
double satpress_   (float *prop, float *t);
double sattemp_    (float *prop, float *p);
double tairsat_    (float *prop, float *hsat, float *fs);
double wetbulb_    (float *prop, float *tdb, float *w, float *fs);
double xiterate_   (float *x0, float *f0, float *x1, float *f1, float *x2, float *f2, int
                   *icount, int *icvg);
double spvol_      (float *prop, float *tdb, float *w, float *fs);;

int    tdb_rh      (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    tdb_twb     (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    tdb_w       (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    tdb_h       (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    tdb_tdp     (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    rh_w        (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat,float *height);
int    w_h         (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    w_twb       (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    rh_h        (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat, float *height);
int    rh_twb      (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat,float *height);
int    h_twb       (float *prop, float *tdb, float *w, float *rh, float *h__, float *twb,
                    float *tdp, float *v, int *errstat,float *height);
```

부록 B. 구성기기 계산 프로그램

1. 팬

가. 압력독립형

(1) 프로그램

```
/******************************************************************************
 *
 *                            PROGRAM FANSIM
 *
 *                                          Filename   : FANSIM.C
 *                                          Developer  : Lee Yong Hee.
 *                                                       Han_il M.E.C
 *                                          CTranslator: Choi Chul Hoon
 *                                          Date       : aug. ,1998
 *
 ******************************************************************************/
/*  Purpose          Calculate the fan power and leaving air temperature     */
/*                   and humidity for fan using simple part load             */
/*                                 characteristics.                          */
/* *************************** Program Descriptions ************************ */
/* Input variables                                                            */
/*          double  M             Dry air mass flow rate (kg/s)               */
/*          double  TEnt              Entering air dry bulb temperature (c)   */
/*          double  WEnt          Entering air humidity ratio                 */
/*                                                                            */
/* Output Variables                                                           */
/*      double  TLvg          Leaving air dry bulb temperature (C)            */
/*      double  WLvg          Leaving air humidity ratio (-)                  */
/*      double  power         Fan power (W)                                   */
/*      double  ErrStat       Error status indicator, 0=ok, 1=error           */
/*                                                                            */
/* Parameters                                                                 */
/*      double  EffMot        Motor drive efficiency                          */
/*      double  MotorLoss     Fraction of motor heat loss to fluid stream     */
/*      double  FlowRated     Rated volumetric flow rate (m3/s)               */
/*      double  Powrated      Rated shaft power (W)                           */
/*      double  PlrContl      Mode for fan control                            */
/*                            PlrContl = 1 : Discharge dampers                */
/*                            PlrContl = 2 : Inlet vanes                      */
/*                            PlrContl = 3 : Variable speed drive             */
/* ========================================================================= */
#include <stdio.h>
#include <math.h>

#include "fanpmp.h"
#include "prop.h"
/* ============================ Local Variables ============================ */
/*          double  effFan          Fan efficiency                            */
/*          double  hEnt            Entering air enthalpy (J/kg)              */
/*          double  rho             Entering moist air density (kg/m3)        */
/*          double  fflp            Fraction of full-load fan power           */
/*          double  plr             Part load flow ratio                      */
/*          double  powShaft        Shaft power (W)                           */
/*          double  qLoss           Heat transfer to fluid stream (W)         */
/*          double  c(i,PlrContl)   Regression Coefficients                   */
/*          double  small           Small number used in place of zero        */
/* ------------------------------------------------------------------------ */
double RHODRY(double *Prop, double TDB, double W);

float prop[19] =
{
     101325.,1006.,4186.,1805.,4186.,1.82e-5,0.00144,0.026,
     0.604 , 998.,2.501e6,287.055,1.,273.15,1.,0.,2093.,-333430.
};
```

```
void main()
{
     int    ErrStat, intPlrContl;
     double Prop[16], TEnt, WEnt, M;
     double P[20], c[4][3];
     double TLvg, WLvg, hLvg, power;
     double hEnt, rho, fflp, plr, powShaft, qLoss, small;
     int    errstat;
     float  tdb, w, rh, h, twb, tdp, v,  height=0.0;

/* Read Input Data */
/* Read_Input(Prop, P, M, TEnt, WEnt); */

/* Input Data */
     M              = 3.4;
     TEnt           = 12.78;
     WEnt           = 0.00835;
     P[PlrContl]    = 3.0;
     P[FlowRated]   = 5.664;
     P[PowRated]    = 17700.0;
     P[EffMot]      = 0.85;
     P[MotorLoss]   = 1.00;
     Prop[Hfg]      = 2501000.0;
     Prop[CpAir]    = 1006.0;
     Prop[CpVap]    = 1805.0;
     Prop[Patm]     = 101325.0;
     Prop[RAir]     = 287.055;
     Prop[TAbsAdd]  = 273.15;


/* Variable, C[][] initializing */
     c[0][0] = 0.3507123;
     c[1][0] = 0.3085;
     c[2][0] = -0.54137;
     c[3][0] = 0.871988;
     c[0][1] = 0.3707;
     c[1][1] = 0.9725;
     c[2][1] = -0.3424;
     c[3][1] = 0.0;
     c[0][2] = 0.00153;
     c[1][2] = 0.005208;
     c[2][2] = 1.1086;
     c[3][2] = -0.11635563;

     small = 1.0E-9;

     ErrStat = 0;

/* Calculate entering moist air properties */
     tdb  = TEnt;
     w    = WEnt;
     tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
     hEnt = h;
     rho = RHODRY(Prop, TEnt, WEnt);

/* Calculate the part load ratio based on rated flow */
     plr = fabs(M) / rho / P[FlowRated];

/* Calculate the fraction of full-load power based on rating point */
/* fflp = C[0] + C[1] * plr + C[2]*plr**2 + C[3]*plr**3 */
/* Regression coefficients, C[i], vary with control mode */
     intPlrContl = (int)P[PlrContl] - 1;
     fflp = c[0][intPlrContl] + plr * (c[1][intPlrContl]
                   + plr * (c[2][intPlrContl] + plr * c[3][intPlrContl]))

/* Calculate the actual fan shaft power and motor power */
     powShaft = P[PowRated] * fflp;
     power    = powShaft / P[EffMot];

/* Calculate the leaving air condition */
/* If flow is zero, ABS[M] < small, the value of M is replaces with */
```

```
/* small of the same sign as M in calculating hLvg */
    qLoss = powShaft + (power - powShaft) * P[MotorLoss];
    hLvg = hEnt + qLoss / ((fabs(M) >= small) ? fabs(M) : small);
    WLvg = WEnt;
    h    = hLvg;
    w    = WLvg;
    w_h(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
    TLvg = tdb;

    printf("TLvg = %f   WLvg = %f    Power = %f\n",TLvg, WLvg, power);
}

/*========================================================================== */
/*      Function    RHODRY(Prop, TDB, W)                                     */
/*-------------------------------------------------------------------------- */
/*      Function    RHODRYB                                                  */
/*      Purpose     Calculate dry air density                                */
/*      Input Variables :                                                    */
/*      Prop                                                                 */
/*      TDW           Dry bulb temperature (C)                               */
/*      W             Humidity ratio (-)                                     */
/*      Return :                                                             */
/*      Rhodry        Density of dry air                                     */
/*      Properties :                                                         */
/*      Patm          Atmoshpere pressure                                    */
/*      Rair          Gas constant for air                                   */
/*      TAbsAdd       Additive constant to user T to absolute T              */
/*-------------------------------------------------------------------------- */
/*========================= Global Variables =============================== */
/*========================= Local Variables ================================ */
/*      pAir          Partial pressure of dry air                            */
/*-------------------------------------------------------------------------- */
double RHODRY(double Prop[], double TDB, double W)
{
    double pAir, RhoDry;

    /* Calculate the dry air density from perfect gas laws */
    pAir = 0.62198 * Prop[Patm] / (0.62198 + W);
    RhoDry = pAir / Prop[RAir] / (TDB + Prop[TAbsAdd]);

    return(RhoDry);
}


/* ========================================================================= */
/* **************************** FANPMP.H ************************************ */
/* ------------------------------------------------------------------------- */
/*      Filename        FANPMP.H                                             */
/*      Purpose         This file is to be include in the detailed fan and  */
/*                      pump routines.  It assigns numbers to the parameter */
/*                      names for the HVAC component.                        */
/*--------------------------------------------------------------------------*/
#define         NPFanPmp        13
#define         EffMot          0
#define         MotorLoss       1
#define         DiamWheel       2
#define         FlowRated       2
#define         PowRated        3
#define         DpRated         4
#define         PlrContl        4
#define         Coef0Fflp       5
#define         Coef1Fflp       6
#define         Coef2Fflp       7
#define         Coef3Fflp       8
#define         Coef0Head       3
#define         Coef1Head       4
#define         Coef2Head       5
#define         Coef3Head       6
#define         Coef4Head       7
#define         Coef0Eff        8
#define         Coef1Eff        9
```

```
#define         Coef2Eff        10
#define         Coef3Eff        11
#define         Coef4Eff        12


/* ====================================================================
 * ******************************** PROP.H ***********************************
 * --------------------------------------------------------------------
 *      Filename PROP.H
 *      Purpose          This file assigns a number to air and water property
 *                       names to be used in the "prop" array
 * --------------------------------------------------------------------
 *     Patm             Atmospheric pressure (Pa)
 *     CpAir            Specific heat of dry air (J/kg C)
 *     CpLiq            Specific heat of saturated water vapor (J/kg C)
 *     DViscAir         Air dynamic viscosity (kg/m s)
 *     DViscLiq         Liquid dynamic viscosity (kg/m s)
 *     KAir             Air thermal conductivity (W/m C)
 *     KLiq             Liquid thermal conductivity (W/m C)
 *     RhoLiq           Liquid density (kg/m3)
 *     Hfg              Latent heat of vaporization of water (J/kg)
 *     RAir             Gas constant for air (J/kg C)
 *     TKelMult         Multiplying factor to convert user T to Kelvin
 *     TAbsAdd          Additive factor to convert user P to kelvin
 *                      tKel = Prop(TKelMult) * T + Prop(TKelAdd)
 *     PaMult           Multiplying factor to convert user P to Pascals
 *     PAbsAdd          Additive factor to convert user P to Pascals
 *                      Pa = Prop(PaMult) * P + Prop(PaAdd)
 * -------------------------------------------------------------------- */
#define         Patm            0
#define         CpAir           1
#define         CpWat           2
#define         CpVap           3
#define         CpLiq           4
#define         DViscAir        5
#define         DViscLiq        6
#define         KAir            7
#define         KLiq            8
#define         RhoLiq          9
#define         Hfg             10
#define         RAir            11
#define         TkelMult        12
#define         TAbsAdd         13
#define         PaMult          14
#define         PAbsAdd         15
```

(2) 계산예제

(가) 입력데이터

건공기유량(M)　　　　: 3.400(kg/s)
입구공기건구온도(TEnt) : 12.780(℃)
입구공기절대습도(WEnt) : 0.00835(-)

(나) 출력

TLvg = 14.335804　WLvg = 0.008350　Power = 5401.194444

나. 압력 종속형

(1) 프로그램

```
/***************************************************************************
 *
 *                       PROGRAM FANDET
 *
 *                                     Filename   : FANDET.C
 *                                     Developer  : Lee Yong Hee
 *                                                  Han_il M.E.C
 *                                     CTranslator: Choi Chul Hoon
 *                                     Date:        Aug., 1998
 *
 ***************************************************************************/
/*
C***********************************************************************
C     SUBROUTINE FANDET(Prop,P,M,PEnt,TEnt,WEnt,Speed,
C    &                  PLvg,TLvg,WLvg,Power,ErrStat)
C***********************************************************************
C*    PURPOSE:              Calculate the performance of a fan given
C*                          the entering fluid conditions and mass
C*                          flow rate.  Results include leaving fluid
C*                          pressure, temperature, and humidity and
C*                          the fan energy consumption.  Calculations
C*                          are based on regression fits to fan head
C*                          and efficiency versus dimensionless flow.
C*                          Uses fan laws as necessary.
C***********************************************************************
C*    INPUT VARIABLES
C*    M             Dry air mass flow rate                        (kg/s)
C*    PEnt          Entering pressure                               (Pa)
C*    TEnt          Entering temperature                             (C)
C*    WEnt          Entering humidity ratio                          (-)
C*    Speed         Rotation speed                                  (Hz)
C*
C*    Note:  If M<0, TEnt and WEnt are assumed to be fan outlet air
C*           conditions, TLvg and WLvg are calculated inlet conditions
C*
C*    OUTPUT VARIABLES
C*    PLvg          Leaving pressure                                (Pa)
C*    TLvg          Leaving temperature                              (C)
C*    WLvg          Leaving humidity ratio                           (-)
C*    Power         Power consumption                                (W)
C*    ErrStat       Error flag, 0 = ok, 1 = error
C*
C*    PARAMETERS
C*    DiamWheel     Diameter of impeller                             (m)
C*    EffMot        Motor drive efficiency                           (-)
C*    MotorLoss     Fraction of motor loss to fluid stream           (-)
C*    Coef(i)Head   Array of coefficients for head vs flow curve
C*       head=coef0+coef1*flow+coef2*flow**2+coef3*flow**3+coef4*flow**4
C*    Coef(i)Eff    Array of coefficients for efficiency curve
C*        eff=coef0+coef1*flow+coef2*flow**2+coef3*flow**3+coef4*flow**4
C*
C*    PROPERTIES
C*    CpAir         Specific heat of dry air                    (J/kg C)
C*    CpVap         Specific heat of water vapor                (J/kg C)
C***********************************************************************
C     INTERNAL VARIABLES:
C     flow          Dimensionless flow coefficient                   (-)
C     head          Dimensionless pressure head coefficient          (-)
C     eff           Fan efficiency                                   (-)
C     hEnt          Entering air enthalpy                         (J/kg)
C     rho           Entering moist air density                   (kg/m3)
C     powShaft      Shaft power                                      (W)
C     qLoss         Heat transfer to fluid stream                    (W)
C     small         Small number used in place of zero
C***********************************************************************/
```

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "fanpmp.h"
#include "prop.h"
#include "psyhead.h"

double sign   (double a,double b);
double RHODRY (double Prop[], double TDB, double W);

float prop[19] =
{
     101325.,1006.,4186.,1805.,4186.,1.82e-5,0.00144,0.026,
     0.604 , 998.,2.501e6,287.055,1.,273.15,1.,0.,2093.,-333430.
};


void main(void)
{
    int   errstat;
    long  _i, _r;
    double deltap, eff, flow, head, hent, hlvg, m, p[NPFanPmp], pent, plvg,
              power, powshaft, qloss, rho, speed, tent, tlvg, went, wlvg;
    double Prop[16];
    double small = 1.e-9;
    float  tdb, w, rh, h, twb, tdp, v,  height=0.0;

    errstat = 0;

    /******* INPUT DATA ****************************** */
    m     = 5.0;
    pent  = 0.0;
    tent  = 25.0;
    went  = 0.010;
    speed = 15.0;

    p[DiamWheel - 1] = 0.760;
    p[EffMot - 1]    = 0.85;
    p[MotorLoss - 1] = 0.0000;
    p[Coef0Head - 1] = 3.46000;
    p[Coef1Head - 1] = 2.28600;
    p[Coef2Head - 1] = -4.10400;
    p[Coef3Head - 1] = -1.53000;
    p[Coef4Head - 1] = 0.000;
    p[Coef0Eff - 1]  = 0.000;
    p[Coef1Eff - 1]  = 2.95700;
    p[Coef2Eff - 1]  = -2.89600;
    p[Coef3Eff - 1]  = 0.2900;
    p[Coef4Eff - 1]  = 0.000;

    Prop[Hfg]      = 2501000.0;
    Prop[CpAir]    = 1006.0;
    Prop[CpVap]    = 1805.0;
    Prop[Patm]     = 101325.0;
    Prop[RAir]     = 287.055;
    Prop[TAbsAdd] = 273.15;

    /*1*** Calculate entering moist air properties */
    tdb  = tent;
    w    = went;
    tdb_w(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
    hent = h;

    rho  = RHODRY( Prop, tent, went );

    /*1*** Calculate the dimensionless flow coefficient */
    flow = fabs( m )/max( rho*speed*pow(p[DiamWheel - 1], 3), small );

    /*1*** Calculate the dimensionless head coefficient and efficiency */
    head = p[Coef0Head - 1] + flow*(p[Coef1Head - 1] + flow*(p[Coef2Head - 1] +
```

```
           flow*(p[Coef3Head - 1] + flow*p[Coef4Head - 1])));
    eff  = p[Coef0Eff - 1] + flow*(p[Coef1Eff - 1] + flow*(p[Coef2Eff - 1] +
           flow*(p[Coef3Eff - 1] + flow*p[Coef4Eff - 1])));

    /*1*** Calculate the power consumption */
    powshaft = fabs( m )*head*pow(speed, 2)*pow(p[DiamWheel - 1], 2)/ max( eff, small )
    power = powshaft/p[EffMot - 1];

    /*1*** Calculate the leaving air conditions
     *2*** If flow is zero, ABS(M) < small, the value of M is replaces with
     *2*** small of the same sign as M in calculating hLvg */

    deltap = head*rho*pow(speed, 2)*pow(p[DiamWheel - 1], 2);
    plvg = pent + sign( max( deltap, small ), m );

    qloss = powshaft + (power - powshaft)*p[MotorLoss - 1];
    hlvg = hent + qloss/sign( max( fabs( m ), small ), m );
    wlvg = went;

    h    = hlvg;
    w    = wlvg;
    w_h(prop, &tdb, &w, &rh, &h, &twb, &tdp, &v, &errstat, &height);
    tlvg = tdb;

    printf(" 1) PLvg (Leaving pressure)       (Pa) =%10.4f\n",plvg);
    printf(" 2) TLvg (Leaving temperature)    (C)  =%10.4f\n",tlvg);
    printf(" 3) WLvg (Leaving humidity ratio) (-)  =%10.4f\n",wlvg);
    printf(" 4) Power(Power consumption)      (W)  =%10.4f\n",power);

    exit(0);
}

/*========================================================================== */
/*      Function    RHODRY(Prop, TDB, W)                                      */
/*-------------------------------------------------------------------------- */
/*      Function    RHODRYB                                                   */
/*      Purpose     Calculate dry air density                                 */
/*      Input Variables :                                                     */
/*      Prop                                                                  */
/*      TDW          Dry bulb temperature (C)                                 */
/*      W            Humidity ratio (-)                                       */
/*      Return :                                                              */
/*      Rhodry       Density of dry air                                       */
/*      Properties :                                                          */
/*      Patm         Atmoshpere pressure                                      */
/*      Rair         Gas constant for air                                     */
/*      TAbsAdd      Additive constant to user T to absolute T                */
/*-------------------------------------------------------------------------- */
double RHODRY(double Prop[], double TDB, double W)
{
    double pAir, RhoDry;

    /* Calculate the dry air density from perfect gas laws */
    pAir = 0.62198 * Prop[Patm] / (0.62198 + W);
    RhoDry = pAir / Prop[RAir] / (TDB + Prop[TAbsAdd]);

    return(RhoDry);
}

double sign(double a,double b)
{
      if(b<0) return -fabs(a);
      else    return  fabs(a);
}
```

(2) 계산예제

(가) 입력데이터

건공기유량(M)	:	5.0(kg/s)
입구공기건구온도(TEnt)	:	25.0(℃)
입구공기절대습도(WEnt)	:	0.01(-)
입구공기압력(PEnt)	:	0.0(Pa)
회전속도(Speed)	:	15.0(Hz)

(나) 출력

1) PLvg (Leaving pressure)	(Pa)	=	421.4599
2) TLvg (Leaving temperature)	(C)	=	25.4544
3) WLvg (Leaving humidity ratio)	(-)	=	0.0100
4) Power(Power consumption)	(W)	=	2737.0740

부록 C. 벽의 열취득 계산 프로그램

1. 프로그램

```
/*****************************************************************************
 *
 *      * TO CALCULATE ---  # 1. Overall Heat Transfer Coefficient
 *                          # 2. RESPONSE FACTOR
 *                          # 3. CONDUCTION TRANSFER FACTOR  --- 7 coef.
 *
 *                                   Filename   : RFCTF.C
 *                                   Developer  : Kim Doo Chun
 *                                                Han_il M.E.C
 *                                   CTranslator: Choi Chul Hoon
 *                                   Date:        Aug. ,1998
 *
 *****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <conio.h>

#define abs(x) ((x) >= 0 ? (x) : -(x))

#define _DEBUG_

#pragma hdrstop

void sub(void);

int  main(int argc, char **argv)
{
     sub();
     return 0;
}
//---------------------------------------------------------------------------

void sub(void)
{
     char   l_unit[10]  = " ( mm )   ";
     char   h_name[12]  = "transfer coe";
     char   r_name[12]  = "    resitance";
     char   u_name[33]  = "Overall heat transfer coefficient";
     char   rf_name[33] = "Response Factor(Triangular Pulse)";
     char   tf_name[46] = "Conduction Transfer Function(Triangular Pulse)";
     char   k_unit[10]  = "(  W/m*K )";
     double c_del       = 1e-6;
     double c_dt        = 1e-6;
     double c_bt        = 1e-6;
     char   h_unit[10]  = "(W/m^2*K)";
     char   r_unit[10]  = "(m^2*K/W)";
     char   u_unit[10]  = "(W/m^2*K)";
     char   um_unit[15] = "(kcal/h*m^2*C)";
     char   l_name[12]  = "    thickness";
     char   k_name[12]  = "conductivity";

     int    i1, i2, i3, i4, i5, i6;
     double d1;
     double beta[300];
     int    menu;
     double ctot, a[900], c[14], d[42];
     int    i, j, k, l, n;
     double r[14], s, y, akmks, btsum, ctsum;
     int    k1;
     double dtsum;
     int    n1;
     double bzsum, czsum;
     int    n2, n3;
     double s1;
```

```
    int    19, n4, n5;
    double w1, w2, w3, w4, cd[56], dd[42];
    int    ia, ib;
    double ak;
    int    ic, id, ie, n20, ig, ij;
    double rc[14], bt[7], ct[7], rf[20], dt[7], bz[7], cz[7], yd;
    int    ni, jx, ix;
    double pn, rt, pm;
    char   matnam[64*14];
    double ww[1200];
    char   walnam[80];
    int    ia1, ib1, ic1, id1, ie1;
    double ww2, ww3, ww4, ww5, ww6, ww7, cdd[56], del, rfb[20];
    int    kim;
    double cmr, rfm[20], tpr[56];

    FILE  *fp_inp,*fp_chk,*fp_out;
    char   title[100]="";
    char   temp[256];

/*****************************************************************************
 *             *** NOMENCLATURE ***
 *                 ------------
 *      * DINENSION
 *                            14 --- Maximum Layers of Composite Wall
 *                           300 --- Maximum Numbers of Roots of B(s)=0.
 *                                 *** Max. of Root --- 1 more over 300.
 *                            10 --- Maxmum Numbers of Response Factors
 *                             7 --- Conduction Transfer Functions being
 *                                   calculated seperately
 *      * CHARACTER              --- For WRITE ( * Unit etc.)
 *  kim ?????????????????????????????????????????????????????????????
 *          Walnam                --- Name of wall or roof : A80
 *          Matnam                --- Name of material of wall/roof : A8
 *                                  ** replace TITLE
 *  kim ?????????????????????????????????????????????????????????????
 *
 *      * INPUT DATA
 *         1. UNIT SYSTEM   : SI
 *         2. INPUT DATA
 *  kim        L9        --- Number of layers
 *  kim        Menu      --- Path controller
 *  kim                     1. Overall heat transfer coefficient
 *  kim                     2. Response factor
 *  kim                     3. Conduction transfer function
 *  kim        Matnam(k) ---  Material Name of the Layer
 *  kim        Tpr(k,1) -(k,4)  ---  Material Dim. and property
 *  kim        Tpr(k,1)              Thickness (mm)
 *  kim        Tpr(k,2)              Conduction : Themal Conductivity(W/m*K)
 *  kim                         Convection : heat transfer coefficent(W/m**2*K)
 *  kim        Tpr(k,3)              Density (kg/m**3)
 *  kim        Tpr(k,4)              Specific heat (kJ/kg*K)
*****************************************************************************
 *         *** Followings must be selected according to the Accuracy.
 *         #1. NRUN --- Number of Runs
 *         #2. C_DEL  --- * Controller of a Number and Magnitude of
 *                          the Characteristic Equation - B(S)=0.
 *                        1. For Research Work    --- 0.1E-08
 *                        2. For Medium      Work --- 0.1E-06
 *                        3. For Engineering Work --- 0.1E-05
 *         #3. C_DT   --- * Controller of a Transrfer Function
 *         #4. C_BT         Coefficient    --- DT(i)    & BT(i)
 *                        1.Research Work --- 0.1E-06  &  0.1E-06
 *                        2.Medium Work   --- 0.1E-06        "
 *                        2.Engineering   --- 0.1E-05        "
*****************************************************************************/

/*        #1. For Research Work */
/*      DATA C_DEL,C_DT,C_BT/0.1E-08,0.1E-06,0.1E-06/ */
/*        #2. For Medium Work */
/*      DATA C_DEL,C_DT,C_BT/0.1E-0          6,0.1E-06,0.1E-06/ */
```

```
/*         #3. For Engineering Work 1 */
    rt   = 0.f;
    ctot = 0.f;

    fp_inp = fopen("rfctf.inp","r+");
//    fp_chk = fopen("rfctf.chk","w+");     !!! check only
//    fp_out = fopen("rfctf.out","w+");

    for(jx = 1; jx <= 2; ++jx)
    {
        for(j = 1; j <= 10; ++j)
        {
            rf[j + jx * 10 - 11]  = 0.f;
            rfb[j + jx * 10 - 11] = 0.f;
        }
    }
    fscanf(fp_inp,"%s\n",&title);
    fscanf(fp_inp,"%d %d\n",&l9,&menu);

    printf("명칭 : %s\n",title);
    if(menu < 2)
    {
       il = 19;
       for(k = 1; k <= il; ++k)
       {
           fscanf(fp_inp,"%s",matnam + (k - 1 << 6));
           for(j = 1; j <= 2; ++j)
           {
               fscanf(fp_inp,"%lf",&tpr[k + j * 14 - 15]);
           }
           // for dummy
           for( j = 1; j <= 2; j++ )
           {
                double dum;
                fscanf(fp_inp,"%lf",&dum);
           }

           if(tpr[k - 1] == (float)0.)  r[k - 1] = (float)1. / tpr[k + 13];
           else                         r[k - 1] = tpr[k - 1] * (float).001 / tpr[k + 13];

           rt += r[k - 1];
       }
    }
    else
    {
       il = 19;

       for(k = 1; k <= il; ++k)
       {
           fscanf(fp_inp,"%s",matnam + (k - 1 << 6));

           for(j = 1; j <= 4; ++j)
           {
               fscanf(fp_inp,"%lf",(char *)&tpr[k + j * 14 - 15]);
           }

           if(tpr[k - 1] == (float)0.)
           {
               r[k - 1] = (float)1. / tpr[k + 13];
               c[k - 1] = (float)0.;
           }
           else
           {
               r[k - 1] = tpr[k - 1] * (float).001 / tpr[k + 13];
               c[k - 1] = tpr[k - 1] * (float).001 * tpr[k + 27] * tpr[k + 41] / (float)3.6;
           }
           rc[k - 1] = r[k - 1] * c[k - 1];
           rt      += r[k - 1];
           ctot    += c[k - 1];
       }
    }
```

```
    ak    = 1.f / rt;
    akmks = ak / 1.163f;

#ifdef _DEBUG_
    printf("층의 수 : %d\n",19);
   if( menu == 1 )
    {
      printf("-------------------------------------------------------------
            -----------------\n");
      printf("Layer Material thickness  conductivity          Resistance \n");
      printf("No.    name      ( mm )     transfer coe(W/m^2*K)  (m^2*K/W) \n");

      printf("-------------------------------------------------------------------
            --------------\n");
      for(k = 1; k <= il; ++k)
      {
         printf("%-4d %-8s ",k,matnam + (k - 1 << 6));
         printf("%9.3g                %9.3g ",tpr[k - 1],tpr[k + 13]);
         printf("%9.3g\n",r[k - 1]);
      }

      printf("-------------------------------------------------------------------
      ----------\n");
      printf("                                   Total = %10.5f\n",rt);
      printf("--------------------------------------------------------------------
      ---------\n");
      printf("                                   K      = %10.5f(W*h/m^2*K)\n",ak);
      printf("                                          = %10.5f(kcal/h*m^2*C)\n",akmks);
      printf("-------------------------------------------------------------------
      -----------\n");
//       getch();
      return;
    }
    else
    {
      printf("--------------------------------------------------------------------
      --------------\n");
      printf("1. 입력 및 중간계산값 \n");
      printf("------------------------------------------------------------------------
      ----------\n");
      printf("                          입력값                                    계산값
            \n");
      printf("---------------------------------------------------------
            ----------------------------\n");
      sprintf(temp,"%3s  %-18s %8s %9s    %9s %9s %9s %9s\n","층","재료명","두께",
            "k or h","밀도","비열","열저항","열용량");
      printf(temp);
      printf("번호                       ( mm )   (W/m*K)     (kg/m^3)
            (kJ/kg*K) (m^2*K/W)(W*h/m^2*K)\n");
      printf("                                  (W/m^2*K)\n");
      printf("---------------------------------------------------------------------
      -------------\n");
      for(k = 1; k <= il; ++k)
      {
         sprintf(temp,"%-4d %-18s %8.3f %9.3f     %9.3f %9.3f %9.5f %9.5f\n",k,matnam +
                (k - 1 << 6),\
         tpr[k+1*14-15],tpr[k+2*14-15],tpr[k+3*14-15],tpr[k+4*14-15],r[k - 1],c[k - 1]);
         printf(temp,"\n");
      }

      printf("------------------------------------------------------------
           ------------------------\n");
      printf("                                                    합계     =   %9.5f
            %9.5f\n",rt, ctot);
      printf("                                                    열관류율 =
            %9.5f(W*h/m^2*K)\n",ak);
      printf("-------------------------------------------------------------------
            -----------------\n");
    }
#endif
```

```
//-------------------------
//   ROOTS OF B(S)=0.
//-------------------------
     i = 0;
L5:
     s = 0.f;
     i1 = 19;

     for(k = 1; k <= i1; ++k)
     {
             d[k - 1]  = 1.f;
             d[k + 13] = r[k - 1];
             d[k + 27] = 0.f;
             dd[k - 1]   = rc[k - 1] * .5f;
             dd[k + 13]  = r[k - 1] * rc[k - 1] / 6.f;
             dd[k + 27]  = c[k - 1];
     }
L7:
     cd[0]   = d[0];
     cd[14]  = d[14];
     cd[28]  = d[28];
     cd[42]  = d[0];
     cdd[0]  = dd[0];
     cdd[14] = dd[14];
     cdd[28] = dd[28];
     cdd[42] = dd[0];
     i1      = 19;

     for(k = 2; k <= i1; ++k)
     {
         k1          = k - 1;
         cd[k - 1]   = cd[k1 - 1] * d[k - 1] + cd[k1 + 13] * d[k + 27];
         cd[k + 13]  = cd[k1 - 1] * d[k + 13] + cd[k1 + 13] * d[k - 1];
         cd[k + 27]  = cd[k1 + 27] * d[k - 1] + cd[k1 + 41] * d[k + 27];
         cd[k + 41]  = cd[k1 + 27] * d[k + 13] + cd[k1 + 41] * d[k - 1];
         cdd[k - 1]  = cd[k1 - 1] * dd[k - 1] + cdd[k1 - 1] * d[k - 1] + cd[k1 + 13] * dd[k +
                  27] + cdd[k1 + 13] * d[k + 27];
         cdd[k + 13] = cd[k1 - 1] * dd[k + 13] + cdd[k1 - 1] * d[k + 13] + cd[k1 + 13] *
                  dd[k - 1] + cdd[k1 + 13] * d[k - 1];
         cdd[k + 27] = cd[k1 + 27] * dd[k - 1] + cdd[k1 + 27] * d[k - 1] + cd[k1 + 41] *
                 dd[k + 27] + cdd[k1 + 41] * d[k + 27];
         cdd[k + 41] = cd[k1 + 27] * dd[k + 13] + cdd[k1 + 27] * d[k + 13] + cd[k1 + 41] *
                 dd[k - 1] + cdd[k1 + 41] * d[k - 1];
     }

     y = cd[19 + 13];
     yd = cdd[19 + 13];
     if(i == 0)
     {
        goto L10;
     }
     w1 = 0.f;
     i1 = i;

     for(ix = 1; ix <= i1; ++ix)
     {
         w3  = s + ww[ix + 299];
         w1 += 1.f / w3;
     }
     yd -= y * w1;
L10:
     s1 = s - y / yd;
     if(s1 >= s)
     {
        goto L12;
     }
     s  = s1;
     i1 = 19;

     for(k = 1; k <= i1; ++k)
```

```
    {
       if(c[k - 1] == 0.f)
       {
               goto L11;
       }
       w1          = sqrt(-s * rc[k - 1]);
       w2          = cos(w1);
       w3          = sin(w1);
       d[k - 1]    = w2;
       d[k + 13]   = r[k - 1] * w3 / w1;
       d[k + 27]   = -w1 * w3 / r[k - 1];
       dd[k - 1]   = rc[k - 1] * .5f * w3 / w1;
       d1          = w1;
       dd[k + 13]  = r[k - 1] * .5f * rc[k - 1] * (w3 / w1 - w2) / (d1 * d1)
       dd[k + 27]  = c[k - 1] * .5f * (w3 / w1 + w2);
L11:
        ;
    }
    goto L7;
L12:
    ++i;
    ww[i + 299] = -s1;
    beta[i - 1] = -s1;

    if(beta[i - 2] < 20.f && beta[i - 1] >= 20.f)
    {
       n20 = i;
    }

    if(ww[i + 299] > 30.f)
    {
       ww[i - 1] = 0.f;
    }
    else
    {
       ww[i - 1] = exp(s1);
    }

    ww[i + 599] = 1.f - ww[i - 1];
    a[i - 1]     = -1.f / s1;
    a[i + 299]   = 1.f / (s1 * cdd[19 + 13]);
    a[i + 599]   = cd[19 + 41] * a[i + 299];
    del = (d1    = a[i + 299] / beta[i - 1], abs(d1));

    if(i - 1 <= 0)
    {
       goto L28;
    }
    else
    {
       goto L27;
    }
L27:
    if(ww[i + 299] - ww[i + 298] <= 0.)
    {
       goto L30;
    }
    else
    {
       goto L28;
    }
L28:
    if(i - 299 <= 0)
    {
       goto L29;
    }
    else
    {
       goto L31;
    }
L29:
```

```
	if(del > c_del)
	{
	   goto L5;
	}
	goto L31;
L30:
	--i;
L31:
	ni   = i;
	il = i;

	for(k = 1; k <= il; ++k)
	{
	   w4 = ww[k + 599] * a[k - 1];
	   w3 = (a[k - 1] - ww[k - 1] / ww[k + 599]) * 2.f;
	   for(jx = 1; jx <= 2; ++jx)
	   {
	       rf[jx * 10 - 10]  += a[k + (jx + 1) * 300 - 301] * w4;
	       rfb[jx * 10 - 10] += w3 * a[k + (jx + 1) * 300 - 301];
	   }
	   ww[k + 299] = w4 * ww[k + 599];
	   ww[k + 599] = ww[k - 1] - w4;
	   ww[k + 899] = 1.f;
	}

	for(j = 2; j <= 10; ++j)
	{
	    il = i;
	    for(k = 1; k <= il; ++k)
	    {
	        for(jx = 1; jx <= 2; ++jx)
	        {
	            w4                     = ww[k + 899] * a[k + (jx + 1) * 300 - 301];
	            rf[j + jx * 10 - 11] -= ww[k + 299] * w4;
	            rfb[j + jx * 10 - 11] += ww[k + 599] * w4;
	        }
	        ww[k + 899] *= ww[k - 1];
	    }
	}

	for(jx = 1; jx <= 2; ++jx)
	{
	    rf[jx * 10 - 10]  += ak;
	    rfb[jx * 10 - 10] += ak;
	}

//	Modified Response factor : RFM(N,JX)                                   !!!
	cmr = exp(-beta[0]);

	for(jx = 1; jx <= 2; ++jx)
	{
	   for(n = 1; n <= 10; ++n)
	   {
	       if(n == 1)
	       {
	           rfm[n + jx * 10 - 11] = rf[n + jx * 10 - 11];
	       }
	       else
	       {
	           rfm[n + jx * 10 - 11] = rf[n + jx * 10 - 11] - cmr * rf[n - 1 + jx * 10 - 11]
	       }
	   }
	}

#ifdef _DEBUG_
	if(menu <= 3)
	{
	   printf("------------------------------------------------------------------
	            ----------------\n");
	   printf("2.특성방정식 B(S)=0의 근 및 계수\n");
	   printf("------------------------------------------------------------------
```

```
                ----------------\n");
        printf("          근                         계수        \n");
        printf("근번호    BETA(i)    관류계수(A(i,2))      흡열계수(A(i,3))\n");
        printf("-----------------------------------------------------------------------
                ----------------\n");

        for(l = 1; l <= 10; ++l)
        {
            il = l - 1;
            printf("%6d %10.5f          %10.5f             %10.5f \n",il,beta[l - 1],a[l +
                  299],a[l + 599]);
        }
        il = ni;
        printf("%6s %10s          %10s             %10s \n",".",".",".",".");
        for(l = ni - 1; l <= il; ++l)
        {
            i2 = l - 1;
            printf("% 6d %10.5f          %10.5f             %10.5f \n",i2,beta[l - 1],a[l +
                  299],a[l + 599]);
        }
        printf("-----------------------------------------------------------------------
                ---------------\n");
    }
    if(menu < 3)
    {
        printf("-----------------------------------------------------------------------
               ---------------\n");
        printf("3.응답계수(이등변 삼각파)\n");
        printf("-----------------------------------------------------------------------
               ---------------\n");
        printf("                                                          Unit :
              %s\n",u_unit);
        printf("-------------------------------------------------------------------
                --------------\n");
        printf("  시계열                     RF                               수정RF\n");
        printf("      n           Y(n)            Z(n)          Ym(n)            Z(n))\n");
        printf("-------------------------------------------------------------------
               --------------\n");
        for(l = 1; l <= 10; ++l)
        {
            i2 = l - 1;
            printf("%8d      %10.5f    %10.5f     %10.5f        %10.5f \n",i2,rf[l - 1],rf[l
                  + 9],rfm[l - 1],rfm[l + 9]);
        }
        printf("-----------------------------------------------------------------------
        -----------\n");
        printf(" 공통비   = %10.5f\n",cmr);
        printf(" 열관류율 = %10.5f(%s)\n",ak,u_unit);
        printf("---------------------------------------------------------------------
        ------------\n");
        return;
    }
#endif

    for(ij = 1; ij <= 7; ++ij)
    {
        dt[ij - 1] = 0.f;
        bt[ij - 1] = 0.f;
        ct[ij - 1] = 0.f;
        bz[ij - 1] = 0.f;
        cz[ij - 1] = 0.f;
    }
    dtsum = 0.f;
    btsum = 0.f;
    ctsum = 0.f;
    bzsum = 0.f;
    czsum = 0.f;
    dt[0] = 1.f;
    i2    = n20;

/*********************************************************************************
```

```
*    TRANSFER FUNCTION --- TRIANGULAR PULSE HOLD
*       * DT  -  COMMON FOR TRIANGULAR AND ZEROTH HOLD
*       * BT & CT
*******************************************************************************/

    for(ia = 1; ia <= i2; ++ia)
    {
        dt[1] -= ww[ia - 1];
    }

    if(n20 < 2)
    {
       goto L777;
    }
    n1   = n20 - 1;
    i2 = n1;

    for(ia = 1; ia <= i2; ++ia)
    {
        ia1  = ia + 1;
        i1 = n20;
        for(ib = ia1; ib <= i1; ++ib)
        {
            dt[2] += ww[ia - 1] * ww[ib - 1];
        }
    }

    if(n20 < 3)
    {
       goto L777;
    }
    n2   = n20 - 2;
    i1 = n2;

    for(ia = 1; ia <= i1; ++ia)
    {
        ia1 = ia + 1;
        i2 = n1;
        for(ib = ia1; ib <= i2; ++ib)
        {
           ib1  = ib + 1;
           i3 = n20;
           for(ic = ib1; ic <= i3; ++ic)
           {
               dt[3] -= ww[ia - 1] * ww[ib - 1] * ww[ic - 1];
           }
        }
    }
    if(n20 < 4)
    {
       goto L777;
    }

    n3   = n20 - 3;
    i3 = n3;

    for(ia = 1; ia <= i3; ++ia)
    {
        ia1  = ia + 1;
        i2 = n2;
        for(ib = ia1; ib <= i2; ++ib)
        {
           ib1  = ib + 1;
           i1 = n1;
           for(ic = ib1; ic <= i1; ++ic)
           {
               ic1  = ic + 1;
               i4 = n20;
               for(id = ic1; id <= i4; ++id)
               {
                   dt[4] += ww[ia - 1] * ww[ib - 1] * ww[ic - 1] * ww[id - 1];
```

```
            }
        }
    }
}

if(n20 < 5)
{
  goto L777;
}
if(abs(dt[4]) < c_dt)
{
   goto L777;
}

n4 = n20 - 4;
i4 = n4;

for(ia = 1; ia <= i4; ++ia)
{
   ia1  = ia + 1;
   i1 = n3;
   for(ib = ia1; ib <= i1; ++ib)
   {
       ib1  = ib + 1;
       i2 = n2;
       for(ic = ib1; ic <= i2; ++ic)
       {
           ic1 = ic + 1;
           i3 = n1;
           for(id = ic1; id <= i3; ++id)
           {
               id1  = id + 1;
               i5 = n20;
               for(ie = id1; ie <= i5; ++ie)
               {
                  dt[5] -= ww[ia - 1] * ww[ib - 1] * ww[ic - 1] * ww[id- 1] * ww[ie - 1]
               }
           }
       }
    }
}

if(n20 < 6)
{
   goto L777;
}
if(abs(dt[5]) < c_dt)
{
   goto L777;
}

n5   = n20 - 5;
i5 = n5;

for(ia = 1; ia <= i5; ++ia)
{
    ia1  = ia + 1;
    i3 = n4;
    for(ib = ia1; ib <= i3; ++ib)
    {
        ib1  = ib + 1;
        i2 = n3;
        for(ic = ib1; ic <= i2; ++ic)
        {
            ic1  = ic + 1;
            i1 = n2;
            for(id = ic1; id <= i1; ++id)
            {
                id1  = id + 1;
                i4 = n1;
                for(ie = id1; ie <= i4; ++ie)
```

```
                    {
                        iel  = ie + 1;
                        i6 = n20;
                        for(ig = iel; ig <= i6; ++ig)
                        {
                            dt[6] += ww[ia - 1] * ww[ib - 1] * ww[ic - 1] * ww[id - 1] * ww[ie
                                    - 1] * ww[ig - 1];
                        }
                    }
                }
            }
        }
    }

L777:

/*******************************************************************************
 *                        BT AND CT COEFFICIENTS ----
 *      ------------------------------------------------
 *          BT(I) --- KIM=1  USING KANRYU COEF.  --- A(I,2)
 *          CT(I) --- KIM=2          HUBYOL      --- A(I,3)
 *******************************************************************************/
    kim = 2;
L99:
    i6 = ni;
    for(j = 1; j <= i6; ++j)
    {
        pn = 1.f - ww[j - 1];
        if(kim == 1)
        {
            pm = pn * a[j + 299] / beta[j - 1];
        }
        else
        {
            pm = pn * a[j + 599] / beta[j - 1];
        }
        ww2 = ww[j - 1] * ww[j - 1];
        ww3 = ww[j - 1] * ww2;
        ww4 = ww2 * ww2;
        ww5 = ww2 * ww3;
        ww6 = ww3 * ww3;
        ww7 = ww4 * ww3;

        bt[0] += pm;
        bt[1] += pm * (dt[1] - pn);
        bt[2] += pm * (dt[2] - dt[1] * pn - ww[j - 1] * pn);
        bt[3] += pm * (dt[3] - dt[2] * pn - dt[1] * ww[j - 1] * pn - ww2 * pn);
        bt[4] += pm * (dt[4] - dt[3] * pn - dt[2] * ww[j - 1] * pn - dt[1] * ww2 * pn - ww3 *
                pn);

       if(abs(bt[4]) < c_bt)
       {
          goto L60;
       }
       bt[5] += pm * (dt[5] - dt[4] * pn - dt[3] * ww[j - 1] * pn - dt[2] * ww2 * pn - dt[1]
              * ww3 * pn - ww4 * pn);

       if(abs(bt[5]) < c_bt)
       {
          goto L60;
       }
        bt[6] += pm * (dt[6] - dt[5] * pn - dt[4] * ww[j - 1] * pn - dt[3] * ww2 * pn - dt[2]
              * ww3 * pn - dt[1] * ww4 * pn - ww5 * pn);
L60:

    }
    bt[0] += ak;
    bt[1] += ak * dt[1];
    bt[2] += ak * dt[2];
    bt[3] += ak * dt[3];
    bt[4] += ak * dt[4];
```

```
     bt[5] += ak * dt[5];
     bt[6] += ak * dt[6];

     if(kim == 1)
     {
        goto L98;
     }
     for(j = 1; j <= 7; ++j)
     {
        ct[j - 1] = bt[j - 1];
     }
     for(j = 1; j <= 7; ++j)
     {
        bt[j - 1] = 0.f;
     }
     --kim;
     goto L99;
L98:

#ifdef _DEBUG_
     for(j = 1; j <= 7; ++j)
     {
        dtsum += dt[j - 1];
        btsum += bt[j - 1];
        ctsum += ct[j - 1];
     }
     printf("-----------------------------------------------------------------
            ---------------\n");
     printf(" 시계열         b(n) (W/m^2*K)            d(n) (-)\n");
     printf("------------------------------------------------------------------
            --------------\n");

     for(n = 1; n <= 7; ++n)
     {
        i6 = n - 1;
        printf(" %2d                 %10.5f            %10.5f\n",i6,bt[n - 1],dt[n - 1])
     }

     printf("--------------------------------------------------------------------
     ------------\n");
     printf("Sum(C(n))   = %10.5f\n",btsum);
     printf("열관류율(U) = %10.3f(W/m^2*K)\n",ak);
     printf("---------------------------------------------------------------------
     -----------\n");
//    getch();
#endif

     return ;
}
```

2. 입력데이터

가. 열관류율 계산 (메뉴번호 = 1)

(1) 입력데이터

```
WALL_1
7  1
실내공기층          0.0            8.3        0.0     0.0
플라스터            3.0            0.740   1300.0     1.1
몰탈               20.0            1.500   2000.0     0.80
보통콘크리트       120.0            1.4     2200.0     0.88
몰탈               25.0            1.5     2000.0     0.80
타일                8.0            1.3     2400.0     0.84
외부공기층          0.0           17.0        0.0     0.0
```

(2) 출력

명칭 : WALL_1
층의 수 : 7

Layer No.	Material name	thickness (mm)	conductivity transfer coe(W/m^2*K)	Resistance (m^2*K/W)
1	실내공기층	0	8.3	0.12
2	플라스터	3	0.74	0.00405
3	몰탈	20	1.5	0.0133
4	보통콘크리트	120	1.4	0.0857
5	몰탈	25	1.5	0.0167
6	타일	8	1.3	0.00615
7	타일	0	17	0.0588
			Total =	0.30523
			K =	3.27624(W*h/m^2*K)
			=	2.81706(kcal/h*m^2*C)

나. 응답계수 계산 (메뉴번호 = 2)

(1) 입력데이터

WALL_1
7 2

실내공기층	0.0	8.3	0.0	0.0
플라스터	3.0	0.740	1300.0	1.1
몰탈	20.0	1.500	2000.0	0.80
보통콘크리트	120.0	1.4	2200.0	0.88
몰탈	25.0	1.5	2000.0	0.80
타일	8.0	1.3	2400.0	0.84
외부공기층	0.0	17.0	0.0	0.0

(2) 출력

명칭 : WALL_1
층의 수 : 7

1. 입력 및 중간계산값

층 번호	재료명	입력값 두께 (mm)	k or h (W/m*K) (W/m^2*K)	밀도 (kg/m^3)	비열 (kJ/kg*K)	계산값 열저항 (m^2*K/W)	열용량 (W*h/m^2*K)
1	실내공기층	0.000	8.300	0.000	0.000	0.12048	0.00000
2	플라스터	3.000	0.740	1300.000	1.100	0.00405	1.19167
3	몰탈	20.000	1.500	2000.000	0.800	0.01333	8.88889
4	보통콘크리트	120.000	1.400	2200.000	0.880	0.08571	64.53334
5	몰탈	25.000	1.500	2000.000	0.800	0.01667	11.11111
6	타일	8.000	1.300	2400.000	0.840	0.00615	4.48000
7	타일	0.000	17.000	0.000	0.000	0.05882	0.00000
				합계	=	0.30523	90.20501
				열관류율	=	3.27624(W*h/m^2*K)	

2. 특성방정식 B(S)=0의 근 및 계수

근번호	근 BETA(i)	계수 관류계수(A(i,2))	흡열계수(A(i,3))

0	0.20739	-4.44011	3.02229
1	1.38624	1.61994	0.85516
2	4.14253	-0.68720	0.33691
3	8.55119	0.36606	0.17806
4	14.64481	-0.21983	0.11137
5	22.53523	0.14207	0.07537
6	32.28883	-0.09862	0.05286
7	43.85395	0.07347	0.03837
8	57.12387	-0.05762	0.02941
9	72.09834	0.04638	0.02420
.	.	.	.
45	1765.99657	0.00185	0.00099
46	1846.24058	-0.00174	0.00091

3.응답계수(이등변 삼각파)

Unit : (W/m^2*K)

시계열 n	RF Y(n)	Z(n)	수정RF Ym(n)	Z(n))
0	0.01158	6.58468	0.01158	6.58468
1	0.22233	-0.97325	0.21292	-6.32462
2	0.44859	-0.50349	0.26791	0.28747
3	0.45502	-0.35937	0.09044	0.04981
4	0.39288	-0.27984	0.02309	0.01222
5	0.32507	-0.22437	0.00578	0.00305
6	0.26563	-0.18159	0.00145	0.00076
7	0.21624	-0.14738	0.00036	0.00019
8	0.17583	-0.11973	0.00009	0.00005
9	0.14292	-0.09729	0.00002	0.00001

공통비 = 0.81270
열관류율 = 3.27624((W/m^2*K))

다. 전도전달함수 계산 (메뉴번호 = 3)

(1) 입력데이터

WALL_1
7 3

실내공기층	0.0	8.3	0.0	0.0
플라스터	3.0	0.740	1300.0	1.1
몰탈	20.0	1.500	2000.0	0.80
보통콘크리트	120.0	1.4	2200.0	0.88
몰탈	25.0	1.5	2000.0	0.80
타일	8.0	1.3	2400.0	0.84
외부공기층	0.0	17.0	0.0	0.0

(2) 출력

명칭 : WALL_1
층의 수 : 7

1. 입력 및 중간계산값

층 번호	재료명	입력값 두께 (mm)	k or h (W/m*K) (W/m^2*K)	밀도 (kg/m^3)	비열 (kJ/kg*K)	계산값 열저항 (m^2*K/W)	열용량 (W*h/m^2*K)
1	실내공기층	0.000	8.300	0.000	0.000	0.12048	0.00000
2	플라스터	3.000	0.740	1300.000	1.100	0.00405	1.19167

3	몰탈	20.000	1.500	2000.000	0.800	0.01333	8.88889
4	보통콘크리트	120.000	1.400	2200.000	0.880	0.08571	64.53334
5	몰탈	25.000	1.500	2000.000	0.800	0.01667	11.11111
6	타일	8.000	1.300	2400.000	0.840	0.00615	4.48000
7	타일	0.000	17.000	0.000	0.000	0.05882	0.00000

합계 = 0.30523 90.20501
열관류율 = 3.27624(W*h/m^2*K)

2.특성방정식 B(S)=0의 근 및 계수

근번호	근 BETA(i)	계수 관류계수(A(i,2))	흡열계수(A(i,3))
0	0.20739	-4.44011	3.02229
1	1.38624	1.61994	0.85516
2	4.14253	-0.68720	0.33691
3	8.55119	0.36606	0.17806
4	14.64481	-0.21983	0.11137
5	22.53523	0.14207	0.07537
6	32.28883	-0.09862	0.05286
7	43.85395	0.07347	0.03837
8	57.12387	-0.05762	0.02941
9	72.09834	0.04638	0.02420
.	.	.	.
45	1765.99657	0.00185	0.00099
46	1846.24058	-0.00174	0.00091

시계열	b(n) (W/m^2*K)	d(n) (-)
0	0.01158	1.00000
1	0.20984	-1.07879
2	0.21130	0.22027
3	0.02001	-0.00327
4	0.00010	0.00000
5	0.00000	0.00000
6	0.00000	0.00000

Sum(C(n)) = 0.45283
열관류율(U) = 3.276(W/m^2*K)

부록 D. 그림자 계산 프로그램

1. 프로그램

```
/******************************************************************************
 *
 *                      PROGRAM SHADOW_CALCULATION
 *
 *                                      Filename   : shadow.cpp
 *                                      Developer  : Kim Doo Chun
 *                                                   Han_il M.E.C
 *                                      CTranslator: Choi Chul Hoon
 *                                      Date:        Aug. ,1998
 *
 ******************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

int   c__1 = 1;

int   lndr_  (int *mon, int *iday);
float fndec_ (float *a);
float fneot_ (float *a);


void submain(void);

int main(int argc, char **argv)
{
    submain();

    return 0;
}

int  lndr_(int *mon, int *iday)
{
     int ret_val;

     ret_val = (*mon - 1) * 30 + (*mon + *mon / 8) / 2 - (*mon + 7) / 10 + *iday;

     return ret_val;
}

float fndec_(float *a)
{
     float ret_val;

     ret_val = (float).006322 - cos(*a + (float).153231) * (float).405748 -
               cos(*a * (float)2. + (float).207099) * (float).00588 - cos(*a * (
               float)3. + (float).620129) * (float).003233;

     return ret_val;
}

float fneot_(float *a)
{
    float ret_val;

    ret_val = cos(*a + (float)1.49831) * (float).122772 - (float)2.79e-4 -
            cos(*a * (float)2. - (float)1.26155) * (float).165458 - cos(*a * (
            float)3. - (float)1.1571) * (float).005354;

    return ret_val;
}

void submain(void)
{
     char *city ="Seoul";
```

```
int   ihr  = 12;
int   iday = 23;
int   mon  = 7;
float x1 = (float)110.;
float x2 = (float)115.;
float x3 = (float)370.;
float x4 = (float)95.;
float x5 = (float)100.;
float y1 = (float)15.;
float y2 = (float)11.;
float y3 = (float)400.;
float y4 = (float)7.;
float y5 = (float)10.;
float alatd = (float)37.34;
float alond = (float)126.58;
float wazid = (float)-90.;
float slopd = (float)90.;

float coef, alat, phid, tdif, tgam;
float salt, wday, tphi, slop, wsaz, gamad;
float awall, awind, saltd, sazid;
float tgamv, xlngh, a1, b1, a3, b3, ylngh, wzrad, wsazd, xwall, tphiv,
          ywall, srwal, srwin, w1, w2, w3, w4, ha, ctheta, thetad, ws,
          totday, dec, cld, sld, eot, snh, yln, saz1, saz2;

printf("\n");
printf("I. INPUT DATA\n");
printf("(1) Location      \n");
printf("        City     = %s  \n  Latitude  (deg)    =%10.3f  \n  Longitude  (deg)  =
       %10.3f\n",city,alatd,alond);
printf("(2) Wall data     \n");
printf("    Azimuth (deg) = %10.2f \nInclination angle (deg) = %10.2f\n",wazid,slopd);
printf("(3) Dimension of Overhang & Side fin\n");
printf("       Side  fin  :  x1  =  %7.1f,  x2  =  %7.1f,x3  =  %7.1f,x4  =  %7.1f,x5  =
      %7.1f\n",x1,x2,x3,x4,x5);
printf("       Overhang  :  y1  =  %7.1f,  y2  =  %7.1f,y3  =  %7.1f,y4  =  %7.1f,y5  =
      %7.1f\n",y1,y2,y3,y4,y5);
printf("(4) Date of calculation\n");
printf("    month = %3d Day   = %3d Hour = %3d \n",mon,iday,ihr);

alat   = alatd * .01745329;
slop   = slopd * .01745329;
totday = (float) lndr_(&mon, &iday);
wday   = totday * 3.1415926 / (float)183.;
dec    = fndec_(&wday);
eot    = fneot_(&wday);
ws     = acos(-tan(alat) * tan(dec));
sld    = sin(alat) * sin(dec);
cld    = cos(alat) * cos(dec);
tdif   = eot + (alond - (float)135.) / (float)15.;

printf("II. CALCULATED RESULT \n");

ha  = ((float) ihr - (float)12. + tdif) * 3.1415926 / (float)12.;
snh = sld + cld * cos(ha);
if(snh <= (float)0.) goto L1028;

salt  = asin(snh);
saltd = salt / .01745329;
saz1  = sin(alat) * snh - sin(dec);
saz2  = cos(salt) * cos(alat);
wzrad = saz1 / saz2;

if(wzrad > (float).99999) sazid = (float)0.;
else                      sazid = acos(wzrad) / .01745329;

if(ha < (float)0.) sazid = -sazid;

wsazd = sazid - wazid;

if(wsazd >= (float)180.)      wsazd += (float)-360.;
```

```
 if(wsazd <= (float)-180.)  wsazd += (float)360.;

 wsaz   = wsazd * .01745329;
 ctheta = cos(salt) * cos(wsaz) * sin(slop) + sin(salt) * cos(slop);
 thetad = acos(ctheta) / .01745329;
 tphiv  = tan(salt) / cos(wsaz);
 tgamv  = tan(wsaz);
 coef   = sin(slop) + tphiv * cos(slop);
 tphi   = (-cos(slop) + tphiv * sin(slop)) / coef;
 tgam   = tgamv / coef;
 phid   = atan(tphi) / .01745329;
 gamad  = atan(tgam) / .01745329;

 if(ctheta < (float)0.)
 {
     if(gamad < (float)0.)  gamad += (float)180.;
     else                   gamad += (float)-180.;
 }

 printf("\n");
 printf("(1) Solar Position Altitude (deg.) = %9.2f\n",saltd);
 printf("    Azimuth (deg.)              = %9.2f\n\
             Wall-Solar Azimuth  (deg.) = %9.2f\n\
             Incident Angle (deg.)      = %9.2f\n",sazid,wsazd,thetad)
 printf("(2) Profile Angle  \n");
 printf("    PHId (deg)     = %9.2f\n\
             Tan(PHId)      = %9.3f\n\
             GAMAd(deg)     = %9.2f\n\
             Tan(GAMAd)     = %9.3f\n",phid,tphi,gamad,tgam);

if(ctheta <= (float)0.) goto L1027;

xwall = x2 + x3 + x4;
ywall = y2 + y3 + y4;
awind = x3 * y3;
awall = xwall * ywall - awind;

if(tgam <= (float)0.)
{
   a1 = x4;
   a3 = x2;
   xlngh = -tgam * x1;
}
else
{
   a1 = x2;
   a3 = x4;
   xlngh = tgam * x5;
}

if(tphi <= (float)0.)
{
   b1 = y4;
   b3 = y2;
   ylngh = -tphi * y5;
}
else
{
   b1 = y2;
   b3 = y4;
   ylngh = tphi * y1;
}


printf("(4) Length of Shadow Cast\n");
printf("    Xlngh = %10.2f Ylngh = %10.2f\n",xlngh,ylngh);

if(xlngh >= xwall || ylngh >= ywall) goto L1023;

w1 = (xwall - xlngh) * (ywall - ylngh);
```

```
    if(xlngh >= x3 + a3) goto L1024;
    if (xlngh <= a3) goto L1019;

    w2 = x3 + a3 - xlngh;

    goto L1020;
L1019:
    w2 = x3;
L1020:
    if (ylngh >= b1 + y3) goto L1024;
    if (ylngh <= b1) goto L1021;
    w3 = (b1 + y3 - ylngh) * w2;
    goto L1022;
L1021:
    w3 = y3 * w2;
L1022:
    w4 = w1 - w3;
    goto L1025;
L1023:
    w3 = (float)0.;
    w4 = (float)0.;
    goto L1025;
L1024:
    w3 = (float)0.;
    w4 = w1;
L1025:
    srwin = w3 / (x3 * y3);
    if(awall == (float)0.)   srwal = (float)0.;
    else                     srwal = w4 / awall;
    goto L1030;
L1028:
    sazid = (float)0.;
    saltd = (float)0.;

    if(ihr <= 12) printf("Before sunrise time !!!\n");
    if(ihr > 12)  printf("After sunset time !!!\n");

    goto L999;
L1027:
    srwin = (float)0.;
    srwal = (float)0.;
    printf("Wall & Window are all in shade !!!\n");
    printf("*** Incident angle > 90.0 *** GAMAd (wall-solar azimuth)  > 90.0  or  < -90.0\n")
L1030:
    printf("III. RESULT - Ratio of Sunlit Area \n");
    printf("*** WALL   :  SRwal = %10.3f \n*** WINDOW :  SRwin = %10.3f",srwal,srwin);
L999:
    return;
}
```

2. 계산예제

가. 입력데이터

(1) 지방의 위도와 경도
위치 = Seoul, 위도 = 37분 34도, 경도 = 126분 58도
(2) 벽데이터
방위각 = -90 °, 경사각=90 °
(3) 날개벽과 차양의 치수
날개벽
x1 = 110.0, x2 = 115.0, x3 = 370.0, x4 = 95.0, x5 = 100.0
차양
y1 = 15.0, y2 = 11.0, y3 = 400.0, y4 = 7.0, y5 = 10.0
(4) 계산날짜
7월 23일 12시

나. 출력

```
I. INPUT DATA
(1) Location
    City  = Seoul
    Latitude (deg)  =     37.340
    Longitude (deg) =    126.580
(2) Wall data
    Azimuth (deg)   =     -90.00
    Inclination angle (deg) =      90.00
(3) Dimension of Overhang & Side fin
    Side fin : x1 =    110.0, x2 =    115.0, x3 =   370.0, x4 =     95.0, x5 =   100.0
    Overhang : y1 =     15.0, y2 =     11.0, y3 =   400.0, y4 =      7.0, y5 =    10.0
(4) Date of calculation
    month =   7 Day   =  23 Hour =  12

II. CALCULATED RESULT
(1) Solar Position Altitude (deg.) =      70.78
    Azimuth (deg.)               =    -29.55
    Wall-Solar Azimuth  (deg.) =      60.45
    Incident Angle (deg.)        =     80.66
(2) Profile Angle
    PHId (deg)     =     80.24
    Tan(PHId)      =     5.816
    GAMAd(deg)     =     60.45
    Tan(GAMAd)     =     1.764
(4) Length of Shadow Cast
    Xlngh =      176.40 Ylngh =      87.24

III. RESULT - Ratio of Sunlit Area
*** WALL   :  SRwal =      0.424
*** WINDOW :  SRwin =      0.631
```

부록 E. 회귀분석 프로그램

1. 프로그램

```
C
C        THIS PROGRAM CALCULATES THE CURVE FITTINGS FOR TWO
C                INDEPENDENT VARIABLES AS FOLLOWS :
C
C      BI-LINEAR
C
C            Z= C1 + C2*X + C3*Y
C
C      BI-QUADRATIC
C
C            Z= C1 + C2*X + C3*X**2 + C4*Y + C5*Y**2 + C6*X*Y
C
C
C     VARIABLES
C
C      INPUT
C           N        SIZE OF MATRIX A(NXN, 3 FOR BI-LINEAR, 6 FOR BI-QUADRATIC)
C           NX       COLUMNS OF INPUT DATA ( 3 FOR X,Y,Z)
C           M        COLUMNS OF SOLUTION VECTOR( 1 FOR THIS PROGRAM)
C           NPT      NUMBER OF DATA POINTS(MUST LARGER THAN N)
C           QJB      JOB NAME  BI-LINEAR OR BI-QUADRATIC
C           DUMMY    CONTROL CHARACTER( END FOR CLOSE INPUT FILE)
C           D(1,NPT) INPUT INDEPENDENT VARIABLE X VALUE
C           D(2,NPT) INPUT INDEPENDENT VARIABLE Y VALUE
C           D(3,NPT) INPUT DEPENDENT VARIABLE Z VALUE
C
C           NP        MAXIMUM RAW SIZE
C           MP        MAXIMUM COLUMN SIZE
C           A(I,J)      NXN MATRIX
C           B(I,M)    ORIGINAL SOLUTION VECTOR
C                   T(I,M)          CALCULATED SOLUTION VECTOR
C           AI(I,J)   INVERSE MATRIX OF A
C           X(I,M)    COEFFICIENTS FOR EQUATION
C           S(I,J)    SUMMATION OF ELEMENT FOR MATRIX A
C           U(K,L)    UNIT MATRIX
C           C(NPT)    CALCULATED Z VALUE
C           E(NPT)    SQUARE OF DIFFERENCES BETWEEN CALCULATED Z AND INPUT Z
C           SE        STANDARD ERROR
C
      PROGRAM BICURVE
C     driver for routine gaussj
      INTEGER MP,NP
      PARAMETER(MP=20,NP=20)
      REAL a(NP,NP),b(NP,MP),ai(NP,NP),x(NP,MP),s(np,np)
      REAL u(NP,NP),t(NP,MP),d(np,np),e(np),c(np)
      CHARACTER qjb*8,dummy*3
c
      open(7,file='BIQUAD1.INP',status='old')
      open(6,file='BIQUAD1.OUT',status='new')
c
10    read(7,'(a)') dummy
      if (dummy.eq.'END') goto 99
      read(7,*) n,nx,m,npt,qjb
c
      write(6,201)n,nx,m,npt,qjb
201   format(5x,'NxN matrix =',i5/5x,'no. of columns of data ='
     1 ,i5/5x,'no. of columns for solution vector =',i5/5x,
     2'no. of data points =',i5/5x,'type of equation = ',a8//)
c
      read(7,*) ((d(k,1), 1=1,nx), k=1,npt)
      if(qjb.eq.'BILINEAR')go to 41
      if(qjb.eq.'BIQUAD1 ')go to 42
c
      write(6,202)
202   format(/2x, 'input error')
      stop
```

```
c
41    continue
      write(6, 203)
203   format(// 5x, ' equation form '//10x, 'Z = c(1) + c(2)*X + c(3)* Y
     1'////)
c
        do 101 i=1,n
        do 101 j=1,n
        s(i,j)=0.
        b(i,j)=0.
101   continue

c
      do 102 k=1,npt
        s(1,2)=s(1,2)+d(k,1)
        s(1,3)=s(1,3)+d(k,2)
        s(2,2)=s(2,2)+d(k,1)**2.
        s(2,3)=s(2,3)+d(k,1)*d(k,2)
        s(3,3)=s(3,3)+d(k,2)**2.
        b(1,m)=b(1,m)+d(k,3)
        b(2,m)=b(2,m)+d(k,1)*d(k,3)
        b(3,m)=b(3,m)+d(k,2)*d(k,3)
102     continue
        do 103 i=1,n
        do 103 j=1,n
        s(j,i)=S(i,j)
        a(i,j)=s(i,j)
        a(1,1)=float(npt)
103   continue
      go to 47
c
42    continue
      write(6, 204)
204   format(// 5x, ' equation form '//10x, ' Z = C(1) + C(2)*X +  C(3)*
     1X**2'/15x,' + C(4)*Y + C(5)*Y**2 + C(6)*X*Y'////)
c
        do 104 i=1,n
        do 104 j=1,n
        s(i,j)=0.
        b(i,m)=0.
104   continue
c
      do 105 k=1,npt
        s(1,2)=s(1,2)+d(k,1)
        s(1,3)=s(1,3)+d(k,1)**2.
        s(1,4)=s(1,4)+d(k,2)
        s(1,5)=s(1,5)+d(k,2)**2.
        s(1,6)=s(1,6)+d(k,1)*d(k,2)
        s(2,2)=s(2,2)+d(k,1)**2.
        s(2,3)=s(2,3)+d(k,1)**3.
        s(2,4)=s(2,4)+d(k,1)*d(k,2)
        s(2,5)=s(2,5)+d(k,1)*d(k,2)**2.
        s(2,6)=s(2,6)+(d(k,1)**2.)*d(k,2)
        s(3,3)=s(3,3)+d(k,1)**4.
        s(3,4)=s(3,4)+(d(k,1)**2.)*d(k,2)
        s(3,5)=s(3,5)+(d(k,1)**2.)*(d(k,2)**2.)
        s(3,6)=s(3,6)+(d(k,1)**3.)*d(k,2)
        s(4,4)=s(4,4)+d(k,2)**2.
        s(4,5)=s(4,5)+d(k,2)**3.
        s(4,6)=s(4,6)+d(k,1)*d(k,2)**2.
        s(5,5)=s(5,5)+d(k,2)**4.
        s(5,6)=s(5,6)+d(k,1)*d(k,2)**3.
        s(6,6)=s(6,6)+d(k,1)**2.*d(k,2)**2.
        b(1,m)=b(1,m)+d(k,3)
        b(2,m)=b(2,m)+d(k,1)*d(k,3)
        b(3,m)=b(3,m)+d(k,1)**2.*d(k,3)
        b(4,m)=b(4,m)+d(k,2)*d(k,3)
        b(5,m)=b(5,m)+d(k,2)**2.*d(k,3)
        b(6,m)=b(6,m)+d(k,1)*d(k,2)*d(k,3)
105    continue
        do 106 i=1,n
```

```
        do 106 j=1,n
        s(j,i)=s(i,j)
        a(i,j)=s(i,j)
        a(1,1)=float(npt)
106   continue

c
47    continue
C     save Matrices for later testing of results
      do 13 l=1,n
        do 11 k=1,n
          ai(k,l)=a(k,l)
11      continue
        do 12 k=1,m
          x(l,k)=b(l,k)
12    continue
13    continue
c
C     invert Matrix
c
      call gaussj(ai,n,NP,x,m,MP)
c
      do 17 k=1,n
        do 16 l=1,n
          u(k,l)=0.0
          do 15 j=1,n
            u(k,l)=u(k,l)+a(k,j)*ai(j,l)
15        continue
16      continue
17    continue
c      check Vector Solutions
      do 21 l=1,m
        do 19 k=1,n
          t(k,l)=0.0
          do 18 j=1,n
            t(k,l)=t(k,l)+a(k,j)*x(j,l)
18        continue
19      continue
21    continue
      write(6,205)
205   format(       ' Coefficients are'/'*****************'//)
      do 107 kp=1,n
      write(6,206)kp,x(kp,m)
107   continue
206   format(5x,'C(',i3,')   =    ',f12.6)
c
c        to calculate error
c
        do 108 i=1,npt
        e(i)=0.
108      continue
c
      if(qjb.eq.'BIQUAD1 ')goto 78
      do 109 k=1,npt
        c(k)=x(1,m)+x(2,m)*d(k,1)+x(3,m)*d(k,2)
        e(k)=e(k)+(d(k,3)-c(k))**2.
109    continue
      s1=e(npt)/float(npt-3)
        se=sqrt(s1)
        goto 79
78    continue
      do 110 i=1,npt
        c(i)=x(1,m)+x(2,m)*d(i,1)+x(3,m)*d(i,1)**2.+x(4,m)*
     1d(i,2)+x(5,m)*d(i,2)**2.+x(6,m)*d(i,1)*d(i,2)
         e(i)=e(i)+(d(i,3)-c(i))**2.
c
110      continue
      s1=e(npt)/float(npt-6)
        se=sqrt(s1)
79      continue
        write(6,207)
```

```
207   format(///8x,'   x    ',5x, '    y    ',7x,' input  z', 5x,
     1'calculated  z'//)
      do 111 k=1,npt
        write(6,208)d(k,1),d(k,2),d(k,3),c(k)
111    continue
208   format(4e15.4)
      write(6,209)se
209   format(///, 5x, 'STANDARD ERROR IS =    ',E10.4/)
43    stop
99    close(7)
      END
c
      SUBROUTINE gaussj(a,n,np,b,m,mp)
      INTEGER m,mp,n,np,NMAX
      REAL a(np,np),b(np,mp)
      PARAMETER (NMAX=50)
      INTEGER i,icol,irow,j,k,l,ll,indxc(NMAX),indxr(NMAX),ipiv(NMAX)
      REAL big,dum,pivinv
      do 11 j=1,n
        ipiv(j)=0
11    continue
      do 22 i=1,n
        big=0.
        do 13 j=1,n
          if(ipiv(j).ne.1)then
            do 12 k=1,n
              if (ipiv(k).eq.0) then
                if (abs(a(j,k)).ge.big)then
                  big=abs(a(j,k))
                  irow=j
                  icol=k
                endif
              else if (ipiv(k).gt.1) then
                pause 'singular matrix in gaussj'
              endif
12          continue
          endif
13      continue
        ipiv(icol)=ipiv(icol)+1
        if (irow.ne.icol) then
          do 14 l=1,n
            dum=a(irow,l)
            a(irow,l)=a(icol,l)
            a(icol,l)=dum
14        continue
          do 15 l=1,m
            dum=b(irow,l)
            b(irow,l)=b(icol,l)
            b(icol,l)=dum
15        continue
        endif
        indxr(i)=irow
        indxc(i)=icol
        if (a(icol,icol).eq.0.) pause 'singular matrix in gaussj'
        pivinv=1./a(icol,icol)
        a(icol,icol)=1.
        do 16 l=1,n
          a(icol,l)=a(icol,l)*pivinv
16      continue
        do 17 l=1,m
          b(icol,l)=b(icol,l)*pivinv
17      continue
        do 21 ll=1,n
          if(ll.ne.icol)then
            dum=a(ll,icol)
            a(ll,icol)=0.
            do 18 l=1,n
              a(ll,l)=a(ll,l)-a(icol,l)*dum
18          continue
            do 19 l=1,m
              b(ll,l)=b(ll,l)-b(icol,l)*dum
```

```
19          continue
          endif
21      continue
22    continue
      do 24 l=n,1,-1
        if(indxr(l).ne.indxc(l))then
          do 23 k=1,n
            dum=a(k,indxr(l))
            a(k,indxr(l))=a(k,indxc(l))
            a(k,indxc(l))=dum
23        continue
        endif
24    continue
      return
      END
```

2. 계산예제

(가) 입력데이터

```
Size of matrix (NxN)
6,3,1,7,BIQUAD1
1.    1.   21.
1.    2.   46.
1.6   4.   146.
2.    8.   465.
5.    5.   380.
6.    6.   540.
8.    9.  1080.
END

SAMPLE DATA FOR BIQUAD1
6         :  N   SIZE OF MATRIX
3         :  NX  COLUMNS OF DATA
1         :  M   COLUMNS OF SOLUTION VECTOR
7         :  NPT NO OF DATA POINT, MUST LARGER THAN N
BILINEAR  :  QJB JOB NAME
```

(나) 출력

```
NxN matrix =    6
no. of columns of data =    3
no. of columns for solution vector =    1
no. of data points =    7
type of equation = BIQUAD1


 equation form

      Z = C(1) + C(2)*X +  C(3)*X**2
          + C(4)*Y + C(5)*Y**2 + C(6)*X*Y

 Coefficients are
*****************


C( 1)   =       1.459602
C( 2)   =       1.315707
C( 3)   =       3.267852
C( 4)   =       4.167701
C( 5)   =       5.042867
C( 6)   =       5.734606

     x            y          input  z     calculated  z

 .1000E+01     .1000E+01     .2100E+02     .2099E+02
 .1000E+01     .2000E+01     .4600E+02     .4602E+02
```

.1600E+01	.4000E+01	.1460E+03	.1460E+03
.2000E+01	.8000E+01	.4650E+03	.4650E+03
.5000E+01	.5000E+01	.3800E+03	.3800E+03
.6000E+01	.6000E+01	.5400E+03	.5400E+03
.8000E+01	.9000E+01	.1080E+04	.1080E+04

STANDARD ERROR IS =   .9766E-03