工學碩士學位論文

인터넷을 이용한 열처리용 전기로 제어기 개발에 관한 연구

A Study on the Development of the Electric Furnace Controller for Heat Treatment based on the Internet

指導教授 吳 珍 錫

2001년 2월

韓國海洋大學校 大學院

船舶運航시스템工學科 船舶自動化專攻

朴 準 用

本 論文을 朴 準 用의 工學碩士 學位論文으로
認准함

主審 : 工學博士　李 相 泰 印

副審 : 工學博士　盧 彰 注 印

副審 : 工學博士　吳 珍 錫 印

2001年 2月

韓國海洋大學校 大學院

船舶運航시스템工學科 船舶自動化專攻

朴　準　用

A Study on the Development of the Electric Furnace Controller for Heat Treatment based on the Internet

by Park Jun-Yong

Department of Ship Operating System Engineering
The Graduate School of Korea Maritime University
Pusan, Republic of Korea

Abstract

The control based on the internet is just getting started: the first commercial projects appeared less than ten year ago but are now finding widespread application.

In this paper, we have developed the electric furnace controller for heat treatment using the internet.

The internet provides a low-cost and widely-available interface that can make teleoperated resources accessible to anyone with Web Brower.

Clients are connected the system which is electric furnace controller by using remote access watch for condition of electric furnace on real-time and it is controlled by them precisely. The system is consisted of WWW(World Wide Web) server, DB(DataBase) server, control server and controller. They are connected to the internet but the communication between control server to controller are transmitted using RS-232C.

The WWW server algorithm has been designed to include log-in function, furnace watch function, etc. The DB server has been used relational database of 2-tire structure supported SQL(Structured Query Language). And it is constituted four tables that saved process-data. The control server has been designed to base on auto-tuning method algorithm for self-controlling.

The controller that embedded in 8-bit RISCA(Reduced Instruction Set Computer Architecture) chip is consisted of distributed temperature sensing module and electric furnace driver module.

It was confirmed that the system is able to control precision temperature, ramp time and holding time for electric furnace. This controller based on the internet has a variety functions for heat treatment. The validity of these functions is confirmed experimentally.

제 1장 서론

1.1 연구배경

오늘날의 인터넷은 1969년 미국 국방부에서 각 부서의 원활한 데이터 교환을 위한 네트워크인 알파넷(ARPA net)을 만듦으로서 출발하였다. 네트워크의 필요성에 의해서 지속적인 연구와 개발을 하면서 1989년 유럽 분자 물리학 연구소(CERN)에 의해 개발된 WWW(World Wide Web)의 출현으로 인터넷의 사용은 급격히 증가하게 되었다. WWW을 통하여서 ASP, CGI, JAVA 등과 같은 인터넷 프로그램이 개발되면서 하드웨어와 연동된 인터넷 제어기를 개발이 가능해졌다. 인터넷을 이용한 제어기개발은 장소에 구애받지 않고 원격지의 자원을 공유할 수 있고, 제어기를 감시 및 통제할 수 있다.[1,2)]

최근 인터넷 분야의 급속한 기술향상은 미국 등의 선진국을 중심으로 인터넷 제어기의 실용화를 앞당기고 있다. 일부 선진국은 산업기기에 적용된 인터넷 제어기를 상용화하고 있는 추세이다.

본 논문에서는 인터넷상에서도 제어관리가 가능한 고정밀, 고신뢰도의 인터넷을 이용한 열처리용 전기로 제어기에 대하여 연구하고자 한다

연구대상인 열처리용 제어기의 제어사양은 화면제어가 가능해야 하고, 더불어 열처리 대상의 다양한 열처리 요구사항을 만족하도록 제어기를 설계 제작해야 한다. 이를 위해 자동동조기법, 다단계 제어유지기법 등의 열처리 시스템의 제어특성과 인터넷 제어기법을 제어기 설계 과정에 반영하기 위한 열처리용 제어 알고리즘에 관련된 연구를 병행하고자 한다.

인터넷을 이용한 열처리용 전기로 제어기는 미세한 온도차나 소성공정 수에 따라서 상이한 기계적 성질을 얻을 수 있기 때문에 제어시 원격지에서도 고정밀의 온도제어와 다단계의 소성공정이 가능하여야 하고, 우수한 재현성을 확보하므로서 완료된 열처리 데이터를 이용하여 동일한 제어공정을 반복할 수 있어야 한다.

본 논문에서는 선박용 기기 및 산업용 기기의 열처리분야에 활용되는 인터넷을 이용한 열처리용 전기로 제어기에 대하여 연구하고자 한다.

1.2 연구개요

인터넷의 사용이 증가하면서 이것을 이용해 원격지의 한정된 자원을 공유하려는 방법이 많이 시도되고 있다. 인터넷과 WWW 인터페이스를 이용하면 누구나 적은 비용으로 자원을 공유할 수 있기 때문이다.

본 논문에서는 클라이언트(client)들이 원격지에서 인터넷을 이용하여 실시간으로 전기로의 제어상태를 감시하고 고정밀, 고신뢰도의 열처리용 전기로 제어 시스템을 구축하려고 한다. 먼저 활발한 연구가 이루어지고 있는 네트웍 시간 지연에 대해 고찰하고 개선알고리즘을 도출한다. 그 알고리즘에 PID제어 중에 자동동조기법을 추가하여 전기로를 제어하는 제어서버(control server)를 구축하고 전기로와 인터페이스 해주는 제어기를 설계 및 제작한다. 제어기는 분산형 온도 감지방식으로 설계하고, 구동부는 최적의 효율로 작동하도록 구성한다. 인터넷이나 제어 서버의 고장으로 인하여 제어가 불가능할 때에는 페일-세이프(fail-safe)로 제어기 자체내에서 작동하도록 설계한다.[3,4,5,6]

WWW 서버로 다수의 클라이언트와 동시에 접속할 수 있는 환경을 구축하고 DB 서버(DataBase server)을 이용하여 2-티어 구조로 제어서버와 연결시킨다.

본 논문은 2장에서는 인터넷의 네트웍 시간지연에 대해 고찰하고 개선알고리즘을 이용하여서 각각의 시스템을 구성한다. 3장에서는 2장을 토대로 하여 실제적으로 시스템을 구축한다. 4장에서는 실험장치에 대하여 설명하고 시스템의 특성을 실험 및 고찰하고 마지막으로 5장에서는 실험을 토대로 결론을 도출한다.

제2장 알고리즘 및 서버 구축

네트워 기술의 발전으로 점점 네트워 속도가 빨라지고 있지만 시간 지연을 전혀 없앨 수는 없다. 이 장에서는 네트워 시간 지연을 고찰하고 개선알고리즘을 도출하여 시스템에 적용시키고 개선알고리즘을 이용하여서 전용망을 구축하고 고정밀, 고신뢰도의 열처리용 전기로 제어시스템을 구축하기 위해 PID제어기를 설계한다.

사용자가 인터넷접속이 가능한 곳에서 접속하여서 제어가 가능하게 해주는 WWW 서버와 데이터를 각 서버와 인터페이스 해주는 DB 서버를 설계하여 인터넷 제어시스템을 구축하고 PID제어기 설계알고리즘을 이용하여서 제어 서버를 설계하여 인터넷 제어시스템에 연동시킨다.

2.1 알고리즘

2.1.1 네트웍 알고리즘

최근, 급속한 인터넷 사용자의 증가로 인하여서 인터넷의 발전은 가속화되고 있으나 반면에 트래픽(traffic)증가로 인하여 접속속도는 감소하고 있는 추세이다. 일반적으로 시간지연은 각각의 DTE(Data Terminal Equipment)에서 신호를 전송할 때 발생한다.

데이터 전송시 시간지연이 T_p인 망에서, DTE A가 시간 0에 전송을 개시하여, T_p 직전시간인 $T_P - \triangle t$ 에 DTE B가 연결된 케이블의 종단근

처까지 신호가 전달된다. 그러나, 이 시간에 DTE B는 현재 캐리어가 아이들(idle)상태이므로, 신호를 전송할 권리가 있다고 판단하여, 신호를 전송하면 두 신호가 충돌된다. DTE B는 이 사실을 감지하고 재밍(jamming)을 전송한 뒤 송신을 중지한다. 이후 2 T_p시간에 충돌사실이 DTE A에 전달되면, DTE A는 자신의 송신을 중지하고, 재전송을 준비한다.

네트워 충돌이 발생한다는 것은 망에 트래픽이 많아 혼잡하다는 것을 의미이다. 따라서, 재전송시 또다시 충돌을 겪는다면, 그 다음의 재전송은 이전의 재전송지연시간 보다 더 긴 시간동안 대기한 후 전송되어야 할 것이다.

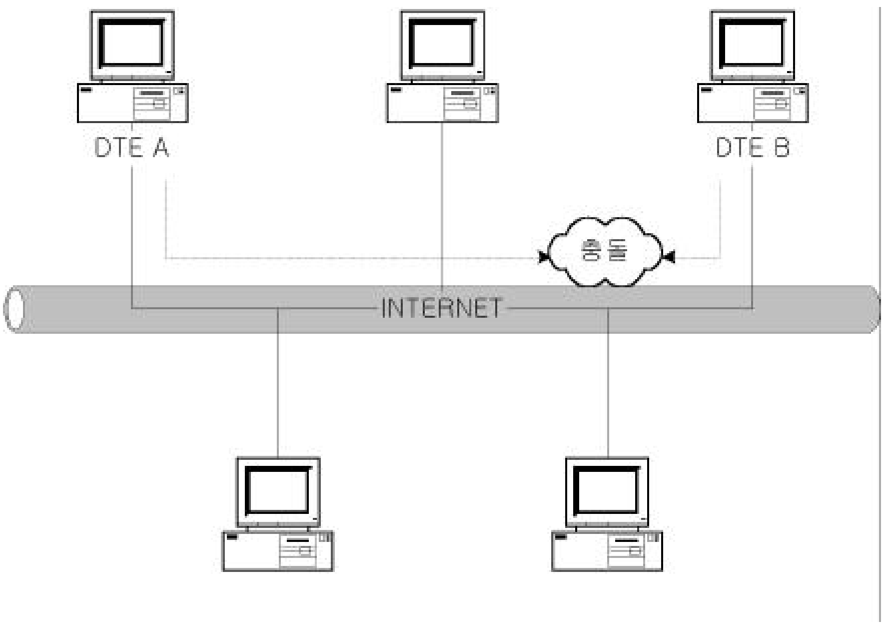

Fig. 2.1 Network transmission delay

앞에서 시간 지연에 대해서 고찰한 내용을 보면 인터넷 사용이 많은 시간에는 트래픽이 더욱 증가하여서 인터넷을 이용한 제어는 불가능할 것이다. 그러므로 전기로와 제어 서버간에는 전용망을 구성하고 사용자와 인터페이스 되는 부분인 WWW 서버와 제어 서버의 자원을 공유하는 DB 서버는 인터넷을 이용한다. 클라이언트와 서버구조는 2-티어구조로 구축한다.[7)]

다수의 클라이언트들이 동시에 접속하여서 감시 및 제어를 할 수 있게 하고 네트워 시간 지연을 고려한 열처리용 전기로 제어 알고리즘은 다음 Fig. 2.2와 같다.

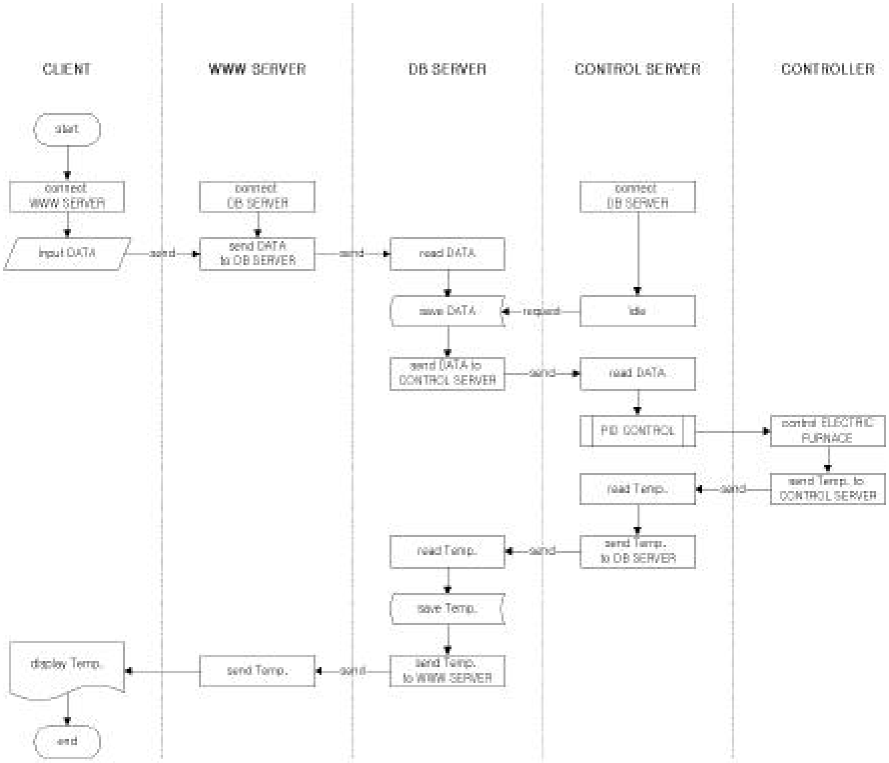

Fig. 2.2 Electric furnace control flowchart for heat treatment

Fig. 2.3은 알고리즘을 바탕으로 설계한 시스템 구성도이다.

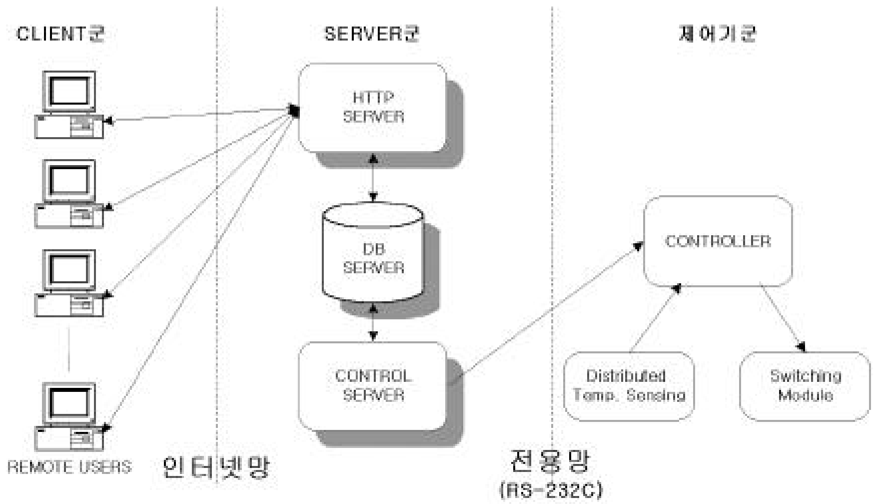

Fig. 2.3 Block diagram of system

2.1.2 전용망 구축 알고리즘

제어기와 제어 서버간에 네트웍 시간 지연을 없애기 위해서 전용망을 사용한다. 전용망으로는 대부분의 컴퓨터나 CPU에 기본사양으로 지원하는 RS-232C를 사용한다. RS-232C는 비평형 회로에 의한 전송회로로서 20K bit/sec 이하의 전송속도를 가지며 장치간의 거리, 즉 전송케이블 길이는 15m이하로 제한되어 있다.

RS-232C는 식(2.1)을 만족하므로 전용망으로 사용할 수 있다.

$$T_t > \frac{T_S - (T_p + T_{ds})}{2} \tag{2.1}$$

위의 식에서 T_t는 제어기와 제어 서버간의 전송시간을 나타낸 것이고, T_S는 제어 서버의 제어연산시 샘플링시간이다. T_p는 제어 서버에서 받은 전송데이터를 처리하는 시간이고, T_{ds}는 분산형 온도 감지 장치에서 열전쌍에서 열기전력을 읽어 선형화하여 CPU에서 처리하는 시간을 나타낸다.[8)]

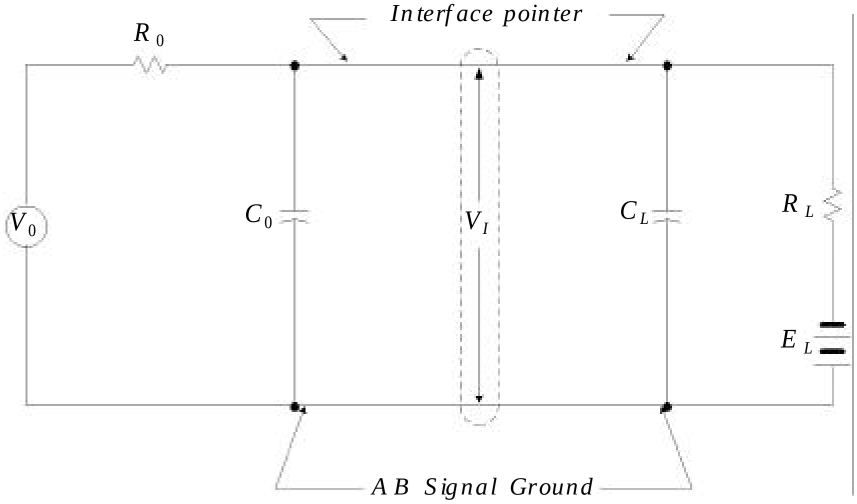

Fig. 2.4 Equivalent circuit of RS-232C

다음 Fig. 2.4는 RS-232C의 등가회로를 나타낸다.

위의 그림에서 V_0은 개방회로 전압, C_0은 케이블의 포함한 접지점에서 측정한 종합실효용량이고 R_0은 회로의 내부직류저항을 나타내고 있다. 그리고 부하측에서 R_1은 직류저항, E_1은 개방전압이고 V_1은 접속점에서 측정한 부하의 종합실효용량이다.

전송망을 구성하기 위해서는 전송라인 설계를 하여야 한다. 전송에서 중

요한 부분은 전류-구동 능력(current-drive capability, i_c)이다. 이것은 전송라인의 용량성 부하 C_t와 전압신호의 상승시간 t에 의해서 구해진다.

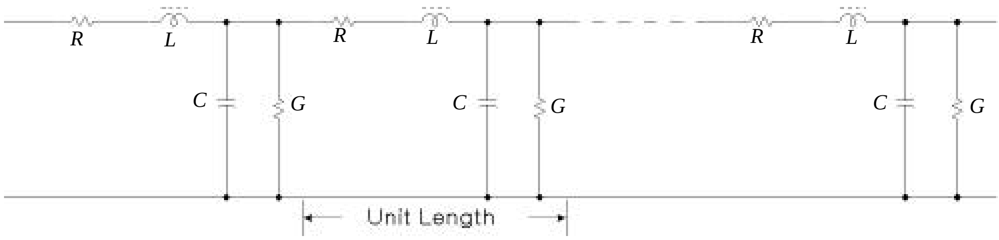

Fig. 2.5 Transmission line model

$$C_t = C \times l \tag{2.2}$$

$$i_c = C_t \frac{dv}{dt} \tag{2.3}$$

여기서 l은 전송라인의 길이이다. 식(2.3)을 만족하는 전송라인을 설계하여야 한다.

전기적 특성중 데이터 신호에서는 부논리, 즉 접속점의 전압이 3V이상일 때는 논리 “0”이고, 3V -3V일 때는 부정, -3V이하일 때는 논리“1”을 사용하고 한다. ±3V를 포함하여 높은 전압범위의 사용은 임피던스가 높은 케이블 사용시 생기는 전기적 노이즈 문제를 최소화시키기 위해서이다.

2.1.3 전기로 제어 알고리즘

열처리용 전기로의 높은 잠열 때문에 응답특성이 매우 느리므로, 샘플링시간(sampling time, T_s)도 일반적인 제어시스템보다 길다. 전기로의 샘플링 주기는 수초에서 수십 분의 시간을 가지는 다양한 주기를 가지고 있다. 그러므로 전기로의 온도공정을 요구특성에 맞게 수행하기 위하여 제어기는 열처리용 전기로의 특성에 맞게 설계해야 한다.

본 논문에서는 제어를 위하여 PID 제어기를 사용하고 지연시간 L을 측정하는 것이 어렵다는 점을 보완하기 위해 자동동조기법(auto tuning method)을 이용한다.

고정밀의 제어를 위해서는 온도 상승시의 제어 파라메타 값과 목표값 T_{SET}에 도달했을 때의 제어 파라메타 값을 각각 구하여서 제어 특성을 향상시키도록 한다.[9)]

자동동조에서 상승시 목표값에 도달하기 전에 선설정값 T_{PRESET}을 정하고 1주기 ON/OFF제어를 실행하여서 온도상승시와 목표값 도달시의 PID 제어파라메타를 계산한다. 전기로 온도제어는 듀티비(duty rate, α)를 이용한다.

선설정값을 구하기 위해서는 α=100%로 하여 온도를 상승시킨다. 상승시 최대기울기 R_S'를 구한다. K값을 실험적으로 구하여 선설정온도값을 얻는다.

$$T_{PRESET} = T_{SET} - K_{PRESET} \cdot T_s \cdot R_s' \tag{2.5}$$

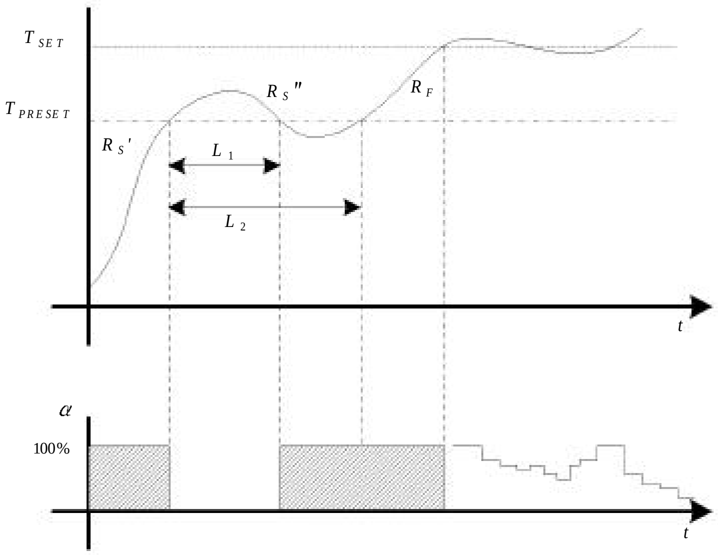

Fig. 2.6 Auto-tuning method controller graph

$$\alpha = \frac{T_{ON}}{T_S} \qquad (2.4)$$

온도가 선설정값에 도달하면 α=0%로 하여 ON/OFF제어를 실시한다. 온도하강시 최대기울기 R_s''를 구한다. 온도 상승시와 하강시의 최대기울기 절대값 중에 큰값을 최대기울기 R_s로 사용한다.

지연시간 L_s은 ON/OFF제어의 최초 반주기의 1/2를 사용한다.

전기로 온도 상승시의 PID 제어파라메타는 지글러·니콜스에 의해서 제안되고 있는 조정법칙에 따라서 다음과 같이 구한다.

$$K_{ps} = \frac{1.2}{R_s L_s}, \quad T_{is} = 2L_s, \quad T_{ds} = 0.5L_s \qquad (2.6)$$

선설정값보다 온도가 하강하면 다시 α=100%로 가열하여 온도가 상승

했을 때 최대기울기 R_F로 사용하고 ON/OFF제어 1주기 L_2의 1/4를 지연시간 L_F로 사용한다.

$$K_{pF} = \frac{1.2}{R_{Fs}L_F}, \quad T_{iF} = 2L_F, \quad T_{dF} = 0.5L_F \qquad (2.7)$$

목표값에 도달했을 때 PID제어가 시작되는 시점을 식(2.8)으로 구한다. K_{SET}은 실험적으로 구한 값이다.

$$T_{PID} = T_{SET} - K_F \cdot T_s \cdot R_F \qquad (2.8)$$

이상의 과정을 통하여 도출한 파라메타를 식(2.9)에 대입하여 원하는 제어량을 구하고 이 값을 이용하여 전기로 온도제어기의 듀티비 α를 조정한다.[10)]

$$U(kT) = K_p \left\{ e(kT) + \frac{T}{T_i}\sum_{i=0}^{k} e(i-1) + \frac{T_d}{T}\left(e(kT) - e((k-1)T)\right) \right\} \qquad (2.9)$$

식(2.9)에서 k=0, 1, 2, …… 이고, $e(kT)$는 오차신호이다.

2.2 서버 구축

2.2.1 WWW 서버

WWW은 인터넷상의 문서를 볼 수 있는 그래픽 형태의 인터페이스를 제공하며 웹에 있는 파일 또는 문서들간에는 하이퍼링크(Hyper Link)의

형태로 구성된 특수문자 또는 그래픽에 의해 서로 연결되어 있다. 페이지는 문자열, 그림, 동영상, 소리 등을 포함시킬 수 있으며 이 페이지는 인터넷이 연결된 세계 각지의 어떤 컴퓨터에도 위치할 수 있다. 하이퍼링크는 내장된 웹 주소를 갖는 문자 또는 그래픽으로써 특정 웹사이트(web site)에 있는 페이지로의 이동이 가능하며 각 페이지는 URL(Uniform Resource Location)이라 불리는 고유의 주소를 갖고 있다. URL은 페이지가 저장되어 있는 컴퓨터의 이름과 그 페이지에 대한 정확한 경로를 지정한다.

WWW 서버는 클라이언트가 접속하여서 전기로 제어기와 인터페이스 할 수 있는 환경을 만들어 준다. 접속하는 사용자에 따라 권한을 부여하여서 그 권한에 따라 전기로를 제어하고 그 결과데이터를 화면에서 감시할 수 있는 기능을 구현한다.

다음 Fig. 2.7은 WWW 서버의 동작 알고리즘을 나타내고 있다.

사용자는 장소에 관계없이 컴퓨터가 인터넷에 연결되어 있으면 WWW 서버의 URL을 이용하여서 접속하면 열처리용 전기로 제어시스템의 WWW 서버와 접속을 할 수 있다. WWW 서버와 접속되면 웹프로그램에 의해서 DB 서버와 접속되어 제어에 필요한 모든 데이터를 동기화 시킨다.

로그인(log-in)은 초기 화면에서 사용자 아이디(user id)와 비밀번호를 입력하면 DB 서버를 검색하여 UDT(UserData Table)과 비교한다. 해당 사용자가 등록된 사용자일 경우 사용자 아이디는 세션(session)객체에 할당된다. 등록된 사용자이면 제어와 감시권한과 열처리 데이터 변경기능을 주고 등록되지 않은 사용자면 손님(guest)권한이 주어져 감시만 할 수 있게 한다. 보안을 위해서 UDT에 저장된 데이터는 제어 서버에서만 입력

및 변경이 가능하도록 하여야 한다.

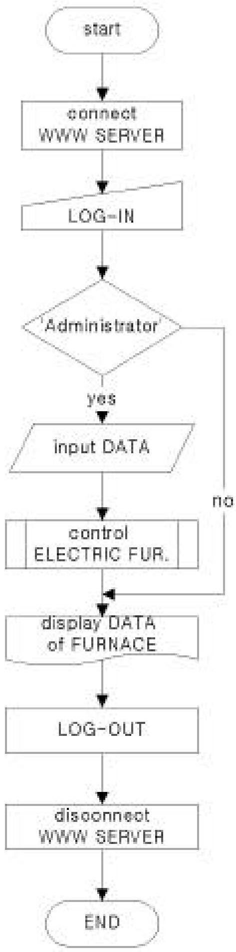

Fig. 2.7 WWW server operation flowchart

열처리 데이터 입력은 등록된 사용자가 WWW 서버를 이용하여서 GUI(Graphic User Interface)로 구성된 화면을 통해 데이터를 입력한다. 입력이 완료되면 DB 서버에 접속하여서 PDT(ProcessData Table)에 데

이터를 업데이트(update)한다.

제어는 입력된 열처리 데이터가 제어 서버로 전송되어서 이를 PID제어 알고리즘을 이용하여서 연산 처리한 후 제어한다. 제어결과를 제어 서버에서는 DB 서버로 전송하여서 이를 WWW 서버에서 읽어들인다. 웹프로그램에서는 사용자가 한눈에 제어상태를 알아볼 수 있게 그래픽을 이용하여서 그래프로 화면에 나타내 준다.

WWW 서버에 사용되는 웹프로그램은 DB 서버와 접속 기능이 있어야 하고 장치 독립적이어야 한다. 그래서 사용자가 사용하는 컴퓨터의 운영체계와 상관없이 동작하여야 한다.

2.2.2 DB 서버

데이터베이스(database)는 Dr. E.F.Codd에 의해 고안된 데이터에 관한 수학적 이론들에 근거하여 데이터를 보다 강력하게 저장하고 복원할 수 있는 시스템을 말한다. 본 시스템에서는 2-티어구조로 구성한다. 2-티어구조에서는 이상적으로 여러 개의 클라이언트에 단일한 프리젠테이션 레이어(presentation layer)를 제공하며, 이것이 중앙에 위치한 데이터 저장 레이어와의 통신을 맡는다. 프리젠테이션 레이어는 보통 클라이언트가 되고, 데이터 저장 레이어는 서버가 된다.[11,12)]

본 논문에서는 자체의 프로그래밍 언어, 인터프리터 및 데이터 형식을 가지며 데이터와 메타데이터를 통합하는 관계 데이터베이스를 사용한다. 관계 데이터베이스는 집합연산을 이용하여 서버 컴퓨터에서 수천 또는 수

백만 개의 레코드를 처리한 다음 아직 작은 결과 집합만을 클라이언트 컴퓨터에 보낼 수 있다. 그러나 하드웨어 요구사항이 엄격하고 서버관리가 난해하다.

WWW 서버와 제어 서버에서 데이터를 검색이 가능하도록 최신 버전의 SQL (Structured Query Language)의 표준인 SQL-92가 사용이 가능하여야 한다.

인터넷을 이용한 열처리용 전기로 제어시스템에 사용될 데이터베이스를 디자인하면 Fig. 2.8과 같다.

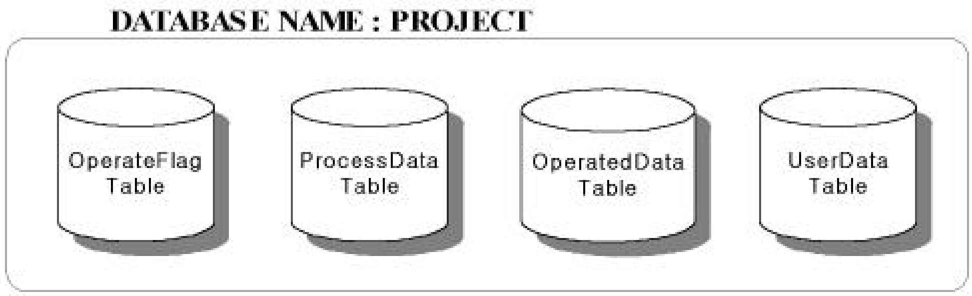

Fig. 2.8 Database structure of control system for heat treatment

데이터베이스의 구조는 4개의 테이블로 구성한다. 4개의 테이블을 구성하고 있는 데이터들이 제어를 하기 위한 공유자원이 되는 것이다.

OFT (OperateFlag Table)은 전기로의 작동과 정지에 관한 플래그(flag)를 저장한다.

PDT (ProcessData Table)은 다단계 제어유지기법을 위한 데이터를 저장한다. 필드(field)는 공정수, 온도, 경사제어시간(ramp control time) 및 유지제어시간(holding control time)으로 구성된다.

ODT (OperatedData Table)은 제어된 데이터를 실시간으로 저장하는 곳이다. 이 테이블을 이용하여서 WWW 서버에서 실시간으로 온도그래프를 구성할 수 있다. 필드는 공정수, 온도, 오차 및 공정단계로 구성된다. 공정단계는 경사제어와 유지제어를 구별하기 위한 필드이다.

마지막으로 UDT (UserData Table)은 전기로를 제어하는 권한을 제한하기 위한 것이다. 클라이언트가 전기로를 제어하기 위해서는 로그온(log-on)을 하여서 권한을 부여받아야 한다. 이 테이블의 필드는 이름, E-메일 및 패스워드(password)로 구성된다.

2.2.3 제어 서버

제어 서버는 앞에서 설계한 제어 알고리즘을 사용하여서 전기로를 제어하고 DB 서버에 자원을 공유시켜주는 역할을 한다. 제어 서버와 제어기 간에는 네트워 시간을 지연을 없애기 위해 전용망인 직렬 통신방식인 RS-232C 통신방식을 이용하고 DB 서버는 지연 시간이 있더라도 실질적인 제어를 하는 것이 아니라 제어 자원을 공유하는 것이므로 인터넷을 사용하여도 무관하다. DB 서버와 연결시 스크립팅언어와 프로그래밍언어에서 호출할 수 있는 이중 인터페이스를 제공하는 ADO(ActiveX Data Object)를 사용하여 데이터 소스 연결, 데이터 소스의 쿼리지정, 쿼리실행 및 오류 탐지하는 수단 제공 등의 일을 수행한다.[13,14,15)]

제어 서버를 구성할 때는 Fig. 2.9에서 보는 모듈형식으로 설계를 한다. 그런 방식으로 프로그램을 작성하려면 추상화(abstraction), 은닉화

(encapsulation), 상속성(inheritance) 및 다형성(polymorphism)의 특징을 가진 객체지향성 프로그래밍(object-oriented programming)방식으로 제작한다.

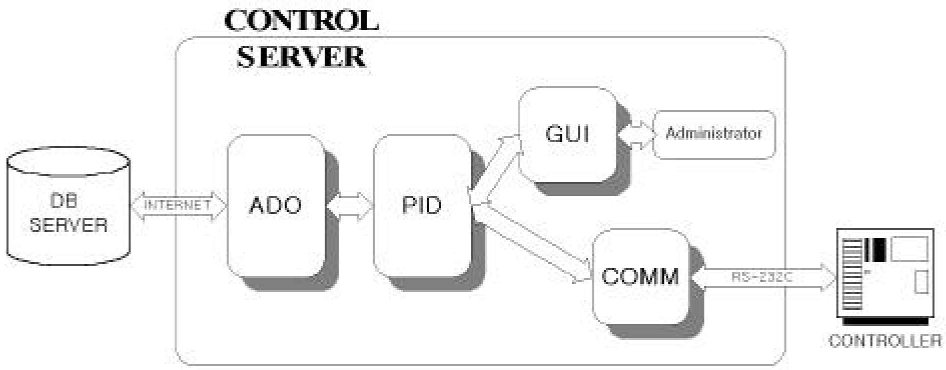

Fig. 2.9 Block diagram of control server

객체지향성 프로그래밍방식이란 기존 방식처럼 순차적으로 프로그램 명령어를 나열해서 프로그램을 작성하는 방법과는 달리, 하나의 독립된 객체로 표현될 수 있는 것들을 하나의 객체로 만들어서 그 안에 그 객체가 가질 수 있는 속성과 자료들 하나로 묶어서 사용함으로써 프로그램의 재사용성(reusability)과 생산성(productivity)을 높인 프로그램방식을 말한다. 객체지향 프로그램방식에서 프로그래밍이란 이러한 객체들을 사용하는 것을 말한다.

ADO를 이용하여서 DB 서버와 연결되면 자원은 동기화가 된다. 서로 동기화된 데이터를 CELEFURView 클래스(CELEFURView class)가 받아서 PID 연산을 하면 그 결과를 CComm 클래스(CCommunication class)로 보내어져 제어기와 통신을 한다.

관리자(administrator)는 GUI기능으로 제어 서버와 인터페이스하여 전기로를 제어 및 감시를 할 수도 있고, 또한 WWW 서버를 통하여서 접속하는 사용자에게 접속권한을 부여할 수 있다. 권한부여는 보안을 위해서 제어 서버에서만 가능하다. 권한부여가 완료되면 DB 서버의 UDT를 업데이트한다.

제3장 시스템 구성

3.1 인터넷 제어시스템 구성

인터넷 제어시스템은 소프트웨어로 구성한 WWW 서버, DB 서버와 제어 서버이다. 이들 서버는 각각 독립적으로 3대의 컴퓨터에 설치할 수 있으나 본 논문에서 처리되는 데이터량이 적어서 데이터 처리속도에 문제가 없으므로 한 대의 컴퓨터에 설치한다. 인터넷 제어시스템을 구성하는 컴퓨터는 정적IP(static)를 사용하고 URL지정시 용이하게 하기 위해서 도메인(domain)을 설정한다. Table.3.1은 인터넷 제어시스템 구성을 나타내는 것이다.

Table. 3.1 Internet control system

HARDWARE	CPU	Intel Pentium Ⅱ 400㎒
	RAM	128MB
	HDD	IBM DNES-309170W(SCSI) 9GB
	LAN CARD	3Com Fast EtherLink XL 10/100Mb TX Ethernet NIC (3C905B-TX)
SOFTWARE	OS	WindowNT4.0 Server(ServicePack5)
	Database	MS-SQL7.0
	WWW	Internet Information Server 4.0
	Program	Java Development Kit Visual Studio 6.0
INTERNET	Web brower	Microsoft Explorer 5.0
	domain	tongmyung.com
	IP	203.255.212.120

3.1.1 DB 설계

본 시스템에서 DB는 각 서버간의 자원을 공유하는 중요한 역할을 담당하고 있다. 그래서 안정적이고 상용화된 마이크로 소프트사의 MS-SQL7.0 서버를 사용한다. MS-SQL은 관리적인 측면에서의 자동화와 분산환경에서의 확장성 그리고 데이터웨어 하우징을 지원하고 관리콘솔에 통합되어 DB 관리자의 부담을 덜어준다.

앞에서 DB 서버를 설계한대로 1개의 DB를 생성하고 그 안에 4개의 테이블(OFT, PDT, OPT, UPT)을 구성한다. 테이블 안에는 제어시스템에 필요한 필드를 구성한다.

먼저 DB 서버의 관리콘솔을 이용하여 project라는 DB를 생성한다. SQL문을 사용하여서 테이블을 생성한다. 생성시 필드에 대한 데이터형, 길이 등에 대한 정보도 같이 첨가한다.

```
CREATE TABLE dbo.OperateFlag(
        state   bit  (1)
)
```

Fig. 3.1 Create OperateFlag Table

```
CREATE TABLE dbo.ProcessData(
        Step  int  (4)  Null,
        SetTemp  int  (4)  Null,
        RampTime  int  (4)  Null,
        HoldTime  int  (4)  Null
)
```

Fig. 3.2 Create ProcessData Table

```
CREATE TABLE dbo.OperatedData(

        Step  int  (4) ,
        NowTemp  int  (4),
        Error  int  (4),
        ProcessStep  int  (4)

)
```

Fig. 3.3 Create OperatedData Table

```
CREATE TABLE dbo.UserData(

        id  varchar  (10),
        passwd varchar  (15),
        email  varchar  (50)

)
```

Fig. 3.4 Create UserData Table

3.1.2 웹 프로그램

본 논문에서 사용되는 웹페이지는 각각의 서버들간에 데이터를 공유하므로 웹페이지에 출력되는 데이터들이 실시간으로 변경되어진다. 그러므로 동적 웹페이지인 ASP(Active Service Page)와 부동소수점연산 및 그래픽처리가 비교적 용이한 자바애플릿(java applet)을 사용하였다.[16)]

ASP는 1995년말에 등장한 마이크로소프트사의 IIS(Internet Information Server)에서 동작하는 웹페이지이다. 기존의 페이지는 서버에 저장된 페이지를 전송함 해주지만 ASP는 사용자에게 보내주기 전에 서버 내에서 연산처리를 하고 그 처리결과를 태그로 만들어 전송하여준다. Fig. 3.5는

ASP연산처리를 나타내는 것이다.

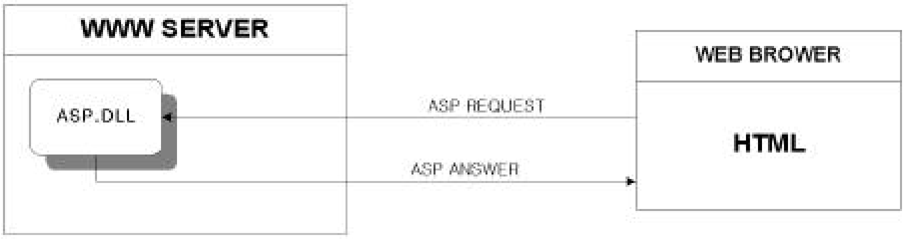

Fig. 3.5 Process ASP

먼저 애플릿프로그래밍이란 WWW에서 사용하는 프로그래밍이다. 이 애플릿을 자바로 구성하는 것이 자바애플릿이다.

자바는 인터넷을 기반으로 만들어진 객체 지향적인 언어로서 클래스에 기초한 프로그래밍언어이다. 클래스는 단순히 객체의 데이터와 메서드(method)의 그룹을 지정하는 데이터 구조이다. 그리고 자바는 인터프리터 형태의 언어와 컴파일러 형태의 언어의 중간 위치에 있다. 자바 프로그램은 간단하지만 고수준의 언어구조를 가지고 있다. 완성된 자바 프로그램 소스코드는 컴파일러를 통해 바이트코드(bytecode)라고 불리는 중간형태의 코드로 컴파일 된다. 이 바이트코드는 자바 런타임 환경(java runtime environment)라고 불리는 인터프리터를 통해 실행되는 것이다. 자바 런타임 환경이 갖춰진 시스템이면 플랫폼에 장치 독립적으로 동작시킬 수 있다.[17)]

부동소수점연산이 가능하고 2차원 그래픽 처리가 용이하다. 애플릿 클래스를 이용하여서 웹브라우저(web brower)에 출력이 가능하다.

웹프로그램은 Fig. 3.6과 같은 형태로 동작하도록 구성한다.

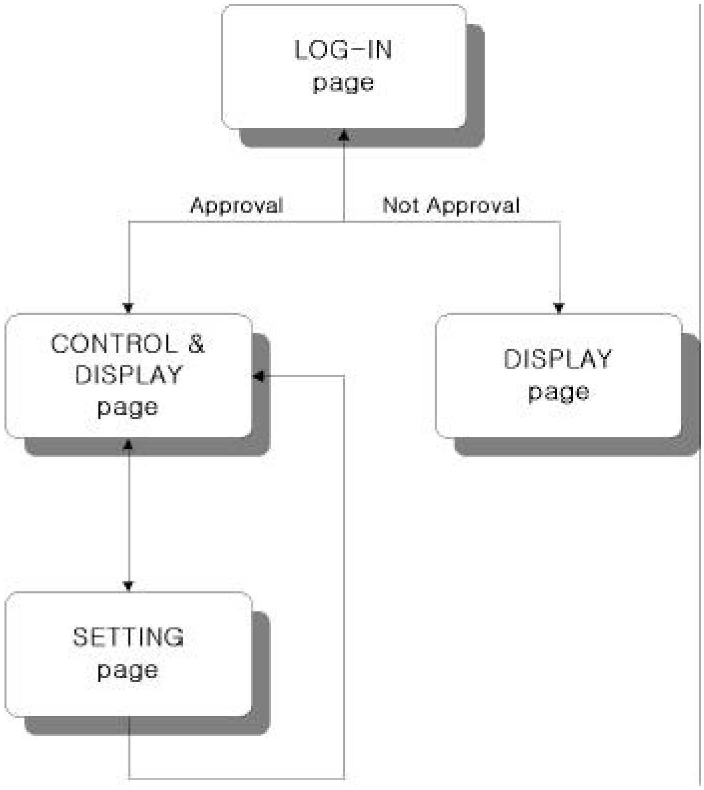

Fig. 3.6 Web program flowchart

3.1.3 제어 프로그램

제어 프로그램은 제어 서버를 구동시키는 프로그램으로서 열처리 데이터를 이용하여서 직접 고정밀, 고신뢰도의 제어를 하는 부분이다.

제어 프로그램은 객체지향성 프로그래밍과 이벤트 처리방식 프로그래밍(Event- Driven Programming)이 가능한 MFC(Microsoft Foundation Class)를 사용하여서 코딩한다.

MFC에는 3가지 형태의 윈도우가 있는데 본 논문에서는 버튼, 스크롤바 등의 액티브엑스(ActiveX)기능을 용이하게 사용할 수 있게 하고 DB 서버와 접속을 원활하게 하기 위해서 CFormView 클래스에서 상속받은 SDI(Single Document Interface)형태의 윈도우를 사용한다.

제어 프로그램에 사용되는 클래스는 Fig. 3.7과 같다.

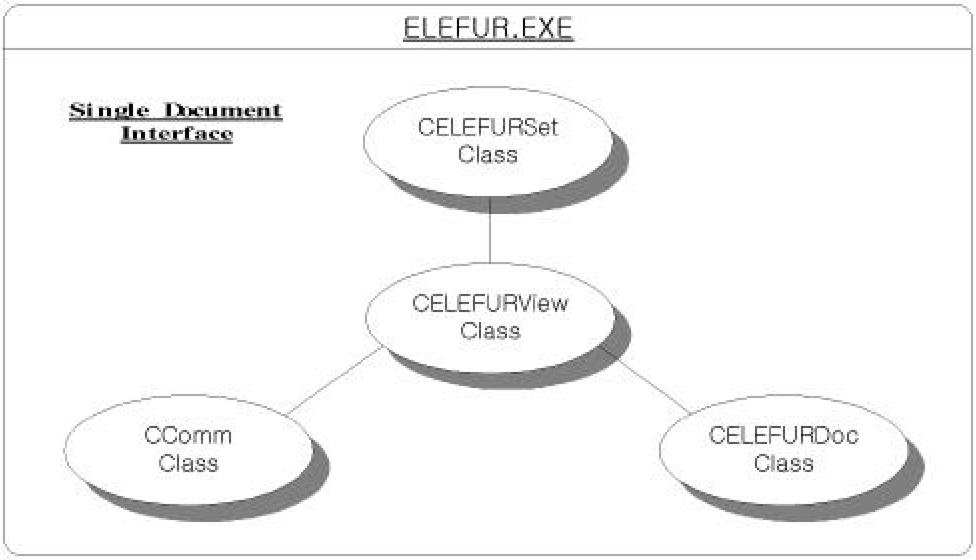

Fig. 3.7 Block diagram of control program class

CELEFURSet 클래스는 열처리용 데이터를 설정하는 콘테이너 윈도우(container window)를 만들어 데이터와 환경설정 변수들을 입력받아서 CELEFURView 클래스에 보내준다.

CComm 클래스는 전용망을 초기화하고 CELEFURSet 클래스의 환경설정 변수들을 가지고 와서 망을 설정하고 제어기와 통신을 하는데 필요한 변수들과 함수들을 정의한다.

```
class   CComm
{
    public:
        //---------- 환경 변수 ------------------------------------------//
        HANDLE          m_hComm;                    // 통신 포트 파일 핸들
        CString         m_sPortName;                // 포트 이름 (COM1 ..)
        BOOL            m_bConnected;               // 포트가 열렸는지 유무를 나타냄.
        OVERLAPPED      m_osRead, m_osWrite;        // 포트 파일 Overlapped structure
        HANDLE          m_hThreadWatchComm;         // Watch함수 Thread 핸들.
        WORD            m_wPortID;                  // WM_COMM_READ와 함께 보내는 인수.

        //---------- 통신 버퍼 ------------------------------------------//
        CQueue          m_QueueRead;

        //---------- 외부 사용 함수 --------------------------------------//
        BOOL            OpenPort(CString sPortName, DWORD dwBaud, WORD wParam);
        void            ClosePort();
        DWORD           WriteComm(BYTE *pBuff, DWORD nToWrite);

        //---------- 내부 사용 함수 --------------------------------------//
        DWORD           ReadComm(BYTE *pBuff, DWORD nToRead);
};
```

Fig. 3.8 Create CComm class

CELEFURDoc 클래스는 CDocument 클래스에서 파생된 클래스로서 SDI 생성시 자동적으로 생성하는 클래스이다. 제어프로그램에서 내부적으로 관리하는 데이터와 함께, 데이터를 조작할 수 있는 멤버함수를 포함하고 있다. 이 프로그램에서는 ADO 컨퍼넌트(ADO component)를 사용하여서 DB 서버와 통신을 담당하는 클래스이다.

CELEFURView 클래스는 제어프로그램에서 물리적인 클라이언트 영역(client area)을 나타내며, 논리적으로 도큐먼트 오브젝트(document object)에 포함된 정보의 뷰포트(viewport)가 된다. 이 클래스에는 설계한 PID 제어기 알고리즘을 소프트웨어로 코딩하였으며 액티브엑스를 이용하여서 화면처리, 이벤트처리 등의 기능을 포함한 클래스이다.

```
class CELEFURDoc : public CDocument
{
public:
	BOOL m_IsConnectionOpen;		// ADO연결여부 나타내는 변수
	_ConnectionPtr m_pConnection;
protected: // create from serialization only
	CELEFURDoc();
	DECLARE_DYNCREATE(CELEFURDoc)
// Overrides
	// ClassWizard generated virtual function overrides
	//{{AFX_VIRTUAL(CELEFURDoc)
	public:
	virtual BOOL OnNewDocument();		// DB 연결시
	virtual void Serialize(CArchive& ar);	// DB 연결해제시
	virtual void OnCloseDocument();
	//}}AFX_VIRTUAL
// Implementation
public:
	virtual ~CELEFURDoc();
#ifdef _DEBUG
	virtual void AssertValid() const;
	virtual void Dump(CDumpContext& dc) const;
#endif
protected:
// Generated message map functions
protected:
	//{{AFX_MSG(CELEFURDoc)
	//}}AFX_MSG
	DECLARE_MESSAGE_MAP()
};
```

Fig. 3.9 Create CELEFURDoc class

```
class CELEFURView : public CFormView
{
protected: // create from serialization only

        --      중략    --

// Attributes
public:          // 인스턴트 생성
        CELEFURDoc* GetDocument();
        CSetDlg         m_dlg;
        CsetTime        m_time;
        CComm           m_pComm;
        CBitmap*        m_pBitmap, *pOldBitmap;
        CDC*            pMemDC;
// Operations
public:
        void PidControl();       // PID 제어기
        void OnTimer(UINT nIDEvent);          // 타이머 함수
        void OnSetPort();        // 전용망 포트설정
        void OnSetSet();         // 열처리 데이터설정
        void OnExit();           // 프로그램 끝내기
        void OnStart();          // 열처리 공정 시작
        void OnEnd();            // 열처리 공정 종료
        int   MaxTemp();         // 최대온도값 구하는 함수
        void SetGraphMode1(int step);         // 공정모드1함수
        void SetGraphMode2(int step);         //공정모드2함수
public:
        virtual ~CELEFURView();
                |
                |
                |
protected:
        //{{AFX_MSG(CELEFURView)
        //}}AFX_MSG
        DECLARE_MESSAGE_MAP()
};
```

Fig. 3.10 Create CELEFURView class

3.2 열처리용 전기로 제어기 구성

3.2.1 개요

제어기는 제어 서버와 전용망으로 구성되어 있어 제어 서버에서 연산한 제어값 $U(kT)$ 을 이용하여서 전기로를 구동시킨다. 그리고 분산형 온도 감지 장치에서 획득한 전기로 내의 온도분포를 감지하여서 제어 서버에 전용망을 통해 전송한다.

이상적인 제어기와 제어 서버간의 구조는 시스템이 오직 특화된 작업만을 처리하도록 하는 것이다. 그래서 제어데이터를 이용하여서 제어 서버는 제어연산을 처리하여 제어기로 전송하고 제어기는 전기로를 구동시키고 온도를 감시하여 제어 서버로 온도데이터를 전송한다. 이런 특화된 작업을 처리하기 위해서는 제어기에는 CPU(Central Processing Unit)가 필요하다. CPU가 통신을 제어하고 제어서버에서 보내온 데이터로 전기로를 구동시키고, 분산형 온도 감지 장치를 제어하여 획득한 온도값을 통신을 통해 제어 서버로 전송한다. 인터넷의 고장이나 제어 서버의 고장으로 인하여 오는 피해를 막기 위해서 페일-세이프(fail-safe)개념을 도입한다. 제어기의 구성은 Fig. 3.11과 같다.[18)]

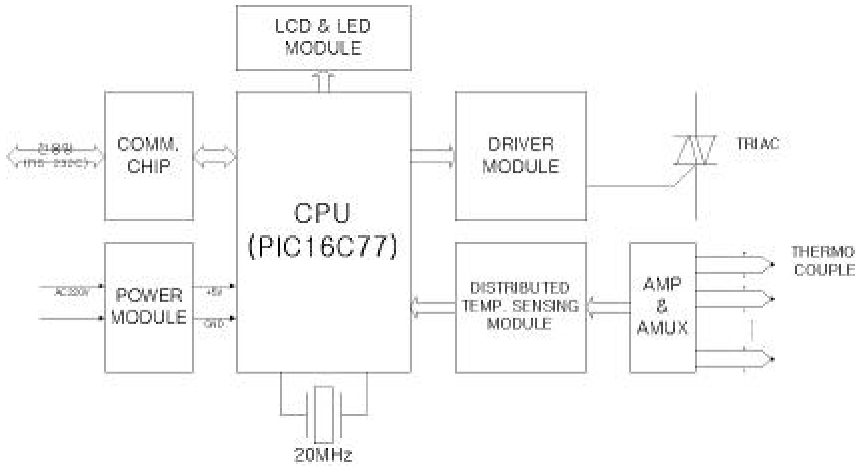

Fig. 3.11 Block diagram of controller

3.2.2 데이터 처리 장치

제어기에서 데이터 처리 장치는 제어 서버와의 통신, 분산형 온도 감지 모듈 제어 및 전기로 스위칭 모듈 제어를 통합 관리한다. 데이터 처리 장치는 CPU, 통신칩 그리고 LCD(Liquid Crystal Display)로 구성한다.

데이터 처리 장치에 사용되는 CPU는 Microchip사의 PIC16C77이다. 이 칩은 파이프라인(pipe line)구조를 가진 RISC 구조(Reduced Instruction Set Computer Architecture)로서 실행시간을 효율적으로 줄일 수 있고 동시에 다음 명령을 패치 할 수 있는 병렬처리가 가능하다. 다음 Table. 3.2는 PIC16C77의 구성을 나타내고 있다.

Table. 3.2 Characteristics of PIC16C77

Program Memory	8MB
Data RAM	192Bytes
Max. Speed	20㎒
I/O	33Ports
ADC(8-Bits)	8
Serial I/O	USART/SPI
PWM	2
Brown-Out Detection	Yes
Timers	3+WDT
In-System Programming	Yes

통신칩은 펌프식 차지(charge)방식을 사용하는 MAXIM사의 MAX232를 사용하고 LCD는 16×2문자를 쓸 수 있는 UC-16206-SNAR5-N을 사용하였다. 데이터 처리 장치를 동작시키는 소프트웨어는 CCS사의 PCM PIC C 컴파일러(compiler)를 사용한다.[19,20] 이 컴파일러의 특징은 부동소수점연산가능, 수학함수 및 표준라이브러리 함수지원, 하드웨어 구동함수 지원, 다양한 에러함수 지원 및 PIC 어셈블러(PIC assembler) 동시사용 가능 등의 특징을 가지고 있다. PCM PIC C 컴파일러의 구성은 다음 Fig. 3.12와 같다. 이 장치를 구동시키는 함수는 Fig. 3.13과 같다.

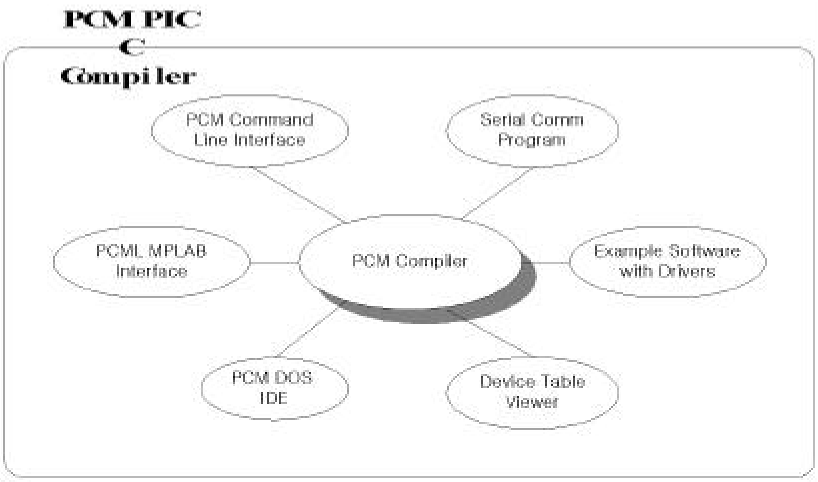

Fig. 3.12 Block diagram of PIC C compiler

ELEFUR.c

```
#include <16c77.h>
#fuses HS, NOPROTECT, NOWDT
#use delay(clock=20000000)            // 사용 clock 정의
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)  // RS-232C 초기화

void init_ext_eeprom();               // 직렬EEPROM 초기화
void lcd_init();                      // LCD 초기화
void DataXmit();                      // 데이터 전송
long NowTemp(char position);          // 온도 검출
void OperateFur(short int state);     // 전기로 작동함수
void DisplayData();                   // 데이터 LCD에 디스플레이
void Control();                       // 제어함수
void enable_interrupts(level);        // 인터럽트 동작
void disable_interrupts(level);       // 인터럽트 정지

void main()
{
            |
}
```

Fig. 3.13 Functions of data process unit

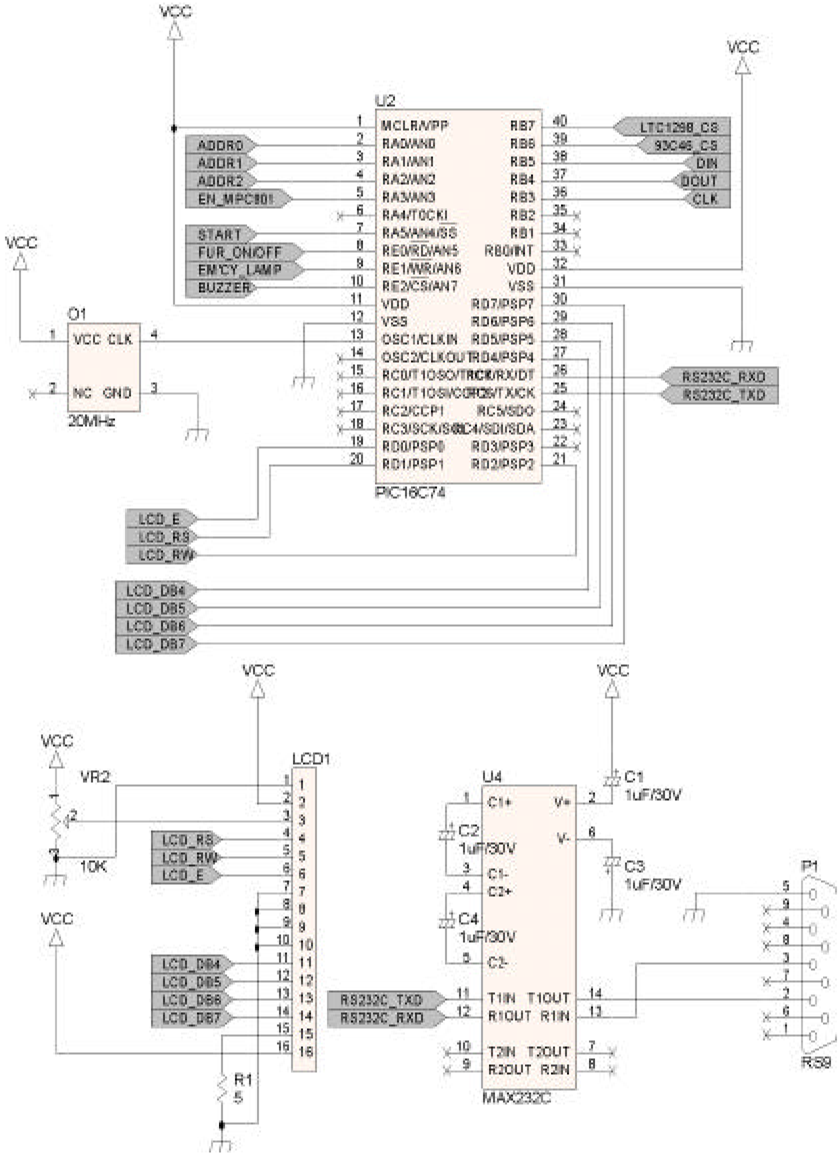

Fig. 3.14 Data process unit circuit

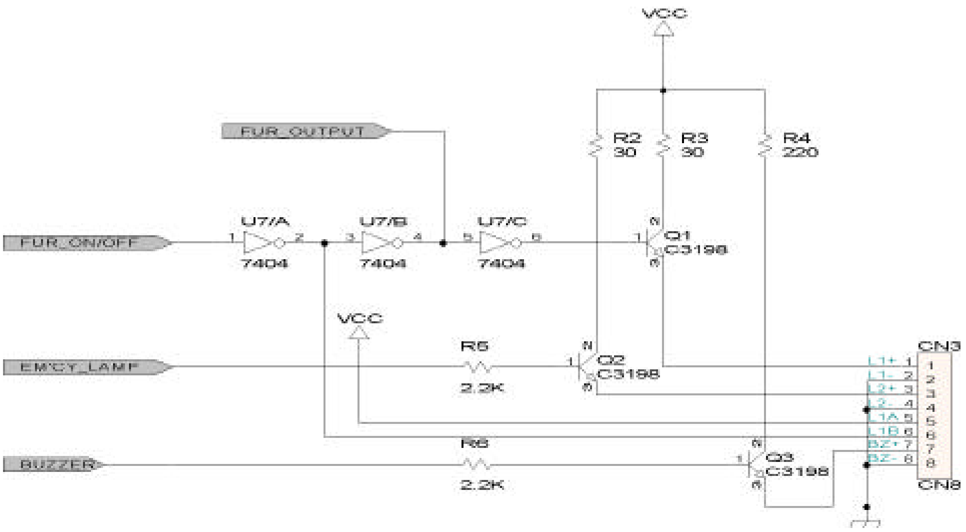

Fig. 3.15 controller I/O circuit

3.2.3 분산형 온도감지 모듈

전기로 내의 열량관계식을 에너지보존법칙에 의존하여 정리하면,

$$Q_e = Q_l + Q_s \tag{3.1}$$

$$Q_l = C\frac{dT_F}{dt} \tag{3.2}$$

$$Q_s = \frac{T_F - T_a}{R_{FS}} \tag{3.3}$$

이다. 여기서 Q_l은 전기로 내의 물체에 전달되는 열량이고, Q_e는 전기히터에 의해 공급되는 열량이며, C는 전기로 내에 있는 물체의 열용량이다. 전기로 표면을 흐르는 열에 대한 열저항은 R_{FS}로 나타낸다. T_F는 전기로내 온도이고, T_a는 대기온도이다. 위의 식(3.3)에서 보는 바와 같이 열

처리용 전기로에는 전기로 내의 있는 물체의 열용량, 대기온도 등의 왜란의 요소가 많으므로 전기로 내의 열량분포를 감지하는 것이 난해하므로 고정밀제어가 난해하다.

분산형 온도 감지 방식은 전기로 내의 열량분포를 다수의 감지점에서 온도를 검출하여서 정확한 제어를 하기 위한 방식이다.

열처리에 사용되는 온도대역은 200℃부터 1600℃까지 다양하게 분포되어 있다. 그러므로 전기로 제어기에서는 열전쌍 R타입[20]을 사용한다. 제어기에서는 아날로그 값인 열기전력을 12비트 AD컨버터로 디지털 값으로 변환한다. 기준접점측의 온도가 0℃일 때 규정하는 JIS의 열기전력표[20]를 보면 R타입의 열기전력은 비선형이므로 디지털 값으로 변환하면 많은 오차를 포함하게 된다. 그러므로 고정밀의 제어를 하기 위해서는 선형화를 해야한다.[21]

선형화방법에는 하드웨어식과 소프트웨어식이 있는데 하드웨어방식은 설계가 복잡하고 실험적으로 보정하여주는 조건이 많으므로 소프트웨어방식을 사용한다. 이것의 일반적인 방법은 절선근사법과 고차방정식법이 있다. 절선근사법은 오차가 심하고 고차방정식법은 수식유도가 난해하므로 본 논문에서는 변형된 절선근사법을 사용하여 오차를 줄인다.

변형된 절선근사방식은 10℃단위는 데이터베이스를 이용하여서 값을 얻고 그 이하는 절선근사방식을 이용하여서 구한 다음 두 값의 합으로 정확한 온도값을 구한다. PIC16C77을 사용하여서 JIS의 열기전력표를 ROM에 데이터베이스화하고 열전쌍에서 나오는 열기전력 $V_i(x)$을 증폭율 k만큼 증폭한 전압값 $V_o(x)$을 구한다.

$$V_o(x) = k \cdot V_i(x) \qquad (3.4)$$

전압값 $V_o(x)$을 12비트 AD컨버터를 이용하여서 구한 값 $f_{v_o}(x)$을 이용하여서 10℃단위의 온도값 T_a을 데이터베이스를 이용하여서 구한다. 온도변환계수 D는 3859이다.

$$T_a = F_{DB}\left(\frac{f_{v_o}(x)}{D} \times 160\right) \qquad (3.5)$$

10℃이하의 온도값 T_b을 구하기 위해서 기울기 $\triangle d$을 구한다.

$$\triangle d = \frac{F_{DB}(n) - F_{DB}(n-1)}{n-(n-1)} = F_{DB}(n) - F_{DB}(n-1) \qquad (3.6)$$

$$T_b = \triangle d \cdot (f_{v_0}(x) - F_{DB}^{-1}(T_a)) \qquad (3.7)$$

$$T = T_a + T_b \qquad (3.8)$$

식(3.7)을 이용하여서 전기로 내의 온도 T를 구할 수 있다.

Fig. 3.18은 변형된 절선근사방식의 알고리즘을 나타내고 있다.

전기로 내의 온도 분포를 감지하기 위해서 4개의 열전쌍을 설치하였고 4개의 아날로그 값을 얻기 위해서 AMUX(Analog Multiplexor)를 사용해 열기전력을 ADC보낸다.

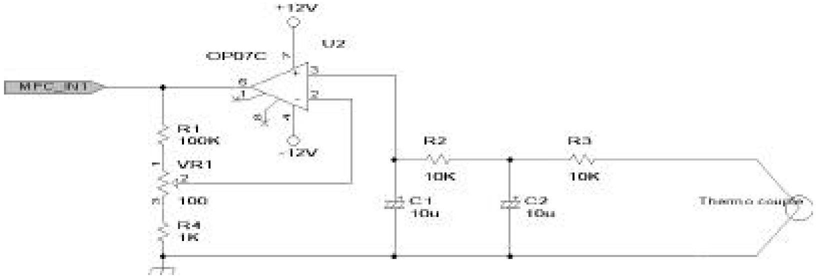

Fig. 3.16 Amp. circuit of thermo couple

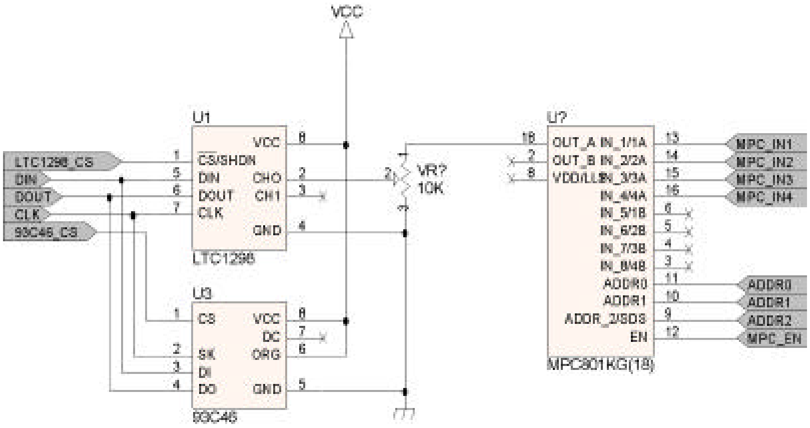

Fig. 3.17 AD converter and Analog Multiplexor circuit

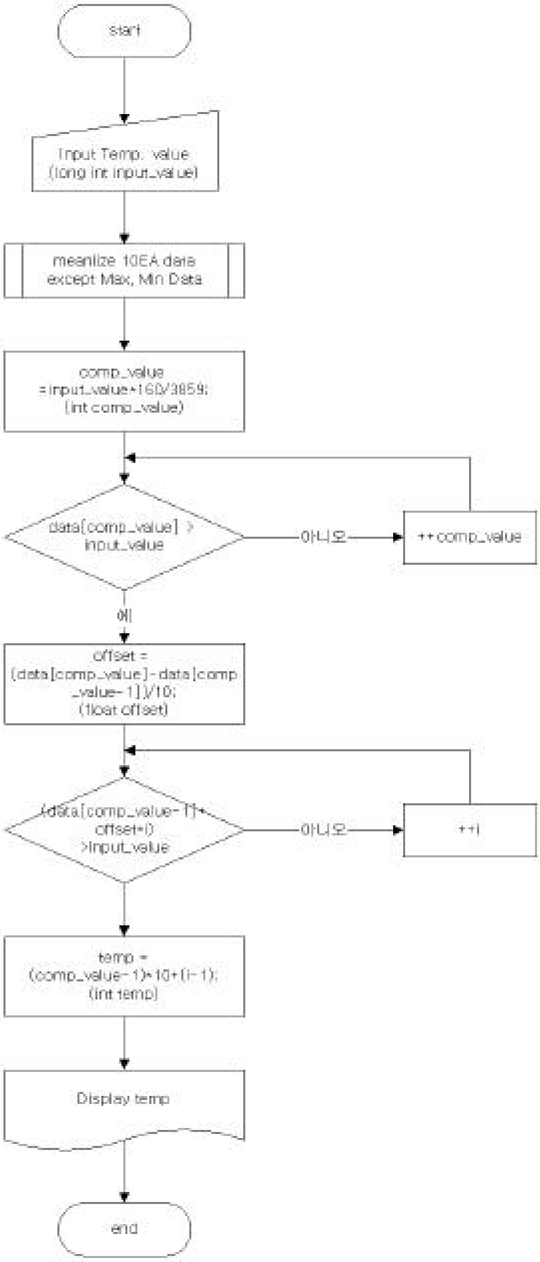

Fig. 3.18 Flowchart of distributed temperature sensing

3.2.4 전기로 스위칭 모듈

전기로를 실질적으로 ON/OFF하는 것이 이 스위칭 모듈이다. 현재 대부분의 전기로에 사용되고 있는 스위칭 소자는 릴레이가 주류를 이룬다. 그러나 응답속도가 느리고 스위칭시 소음과 아크의 발생으로 인하여 고장이 잦다.

빠른 스위칭 속도는 고정밀 전기로 제어의 중요한 요소 중에 하나이므로 본 실험에서는 SanRex사의 TG25C40라는 트라이악(traic)을 사용한다. 이 소자는 무접점스위치소자로서 반영구적으로 사용할 수 있고 스위칭속도가 매우 빠르다. 25A의 전류를 연속적으로 흘릴 수 있고 순간 서지전류(surge current)는 250A까지, 순간 서지전압(surge voltage)은 AC2500V까지 견딜 수 있게 설계되었다. Table. 3.3은 TG25C40에 대한 전기적 특성을 나타내고 있다.

스위칭시 발생하는 노이즈가 데이터 처리 장치에 영향을 최소한으로 억제하기 위해서 MOC3040소자를 이용하여서 광전송시스템을 사용한다.

Table. 3.3 Electrical Characteristics of TG25C40

Item	Ratings	Unit
Repetitive Peak Off-state Current, max	5	mA
Peak On-State Voltage, max	1.4	V
Non-Trigger Gate Voltage, min	0.2	V
Turn-On Time, max	10	V

안정된 전원공급을 위해서 자체적으로 전원장치를 설계 제작하였다. 전

원리플(power ripple)**은 분산형 온도감지 장치에 노이즈가 될 수 있기 때문에 리플감소를 위해서 전압레귤레이터**(voltage regulator)**와 전해콘덴서를 설치하였다**.

Fig. 3.18**은 전기로 스위칭 모듈과 전원공급 모듈의 회로도를 나타내고 있다**.

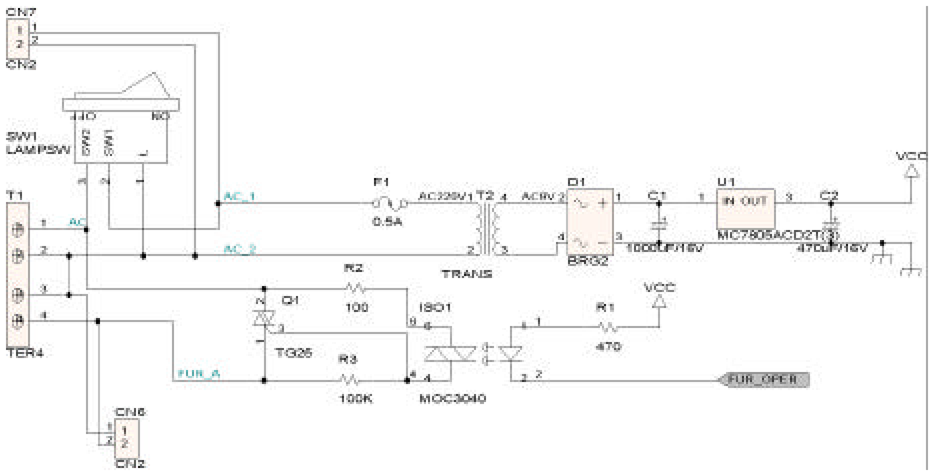

Fig. 3.18 Furnace switching module circuit

제4장 실험 및 고찰

4.1 실험장치

2장과 3장에서 설계한 대로 인터넷을 이용한 열처리용 전기로 제어 시스템을 Fig. 4.1과 같이 제작하였다. 장치(a)는 WWW 서버, DB 서버 및 제어 서버가 설치된 컴퓨터, 장치(b)는 실험용 전기로이고 장치(c)는 전기로 제어기이다. 장치(d)는 제어기 내부에 설치된 CPU 프로그램 개발에 필요한 에뮬레이터(emulator)이다.

Fig. 4.1 Electric furnace controller for heat treatment using Internet

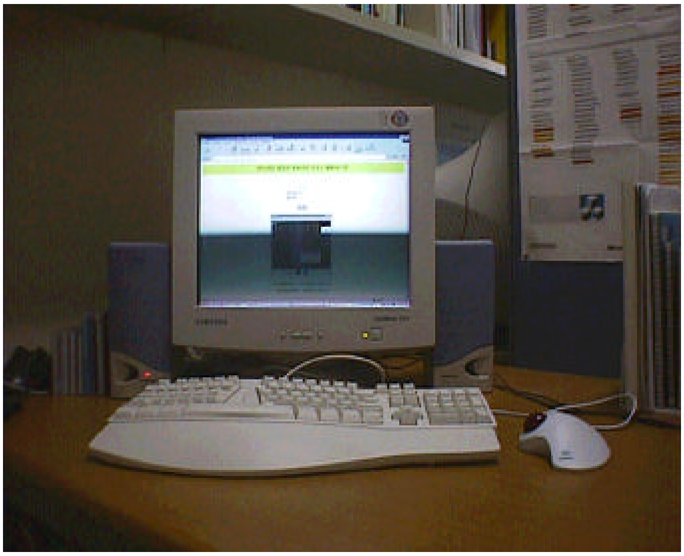

Fig. 4.2 Client computer

Fig. 4.3 External view of controller

Fig. 4.2는 인터넷을 이용하여서 접속한 클라이언트 컴퓨터이다. 이 컴퓨터는 서버 프로그램이 설치된 컴퓨터와 스위치 허브(switch hub)로 연결되었고 정적IP를 사용하였다.

Fig. 4.3은 제어기 외형을 나타내고 Fig. 4.4는 제어기 내부를 나타내고 있다.

Fig. 4.5는 제어기에 부착된 LCD가 제어 상태를 나타내고 있다.

Fig. 4.6은 전기로 내의 온도를 감지하는 R타입 열전쌍이 전기로에 설치된 모습을 나타내고 있다.

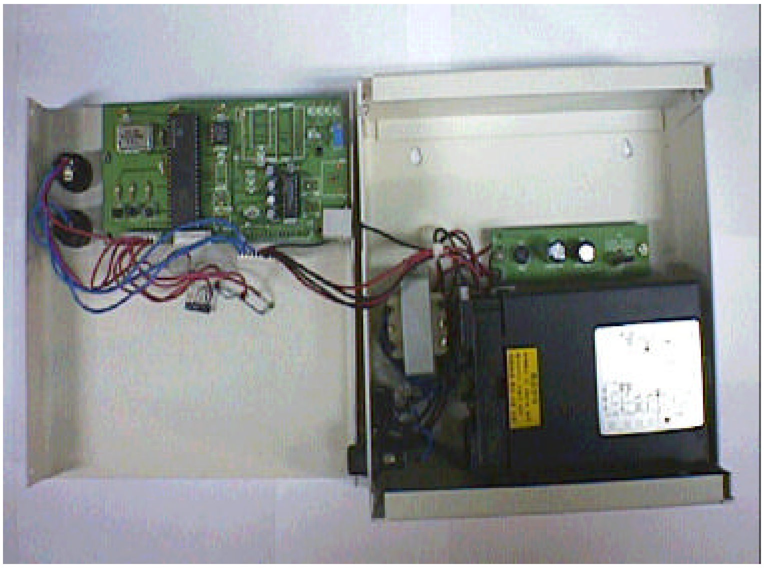

Fig. 4.4 Internal view of controller

Fig. 4.5는 제어기에 부착된 LCD가 제어 상태를 나타내고 있다. Fig. 4.6은 전기로 내에 온도를 감지하는 R타입 열전쌍을 나타내고 있다.

Fig. 4.6 LCD of data process unit

Fig. 4.6 "R"type temperature
sensor

4.2 인터넷 운용 특성 실험

이 실험은 각 서버간의 네트웍 시간 지연을 측정하는 실험이다. 클라이언트가 WWW 서버에 접속하여 DB 서버를 통해 제어 서버로 열처리 데이터를 보낼 때 걸리는 지연 시간과 제어 서버가 DB 서버를 통해서

WWW 서버로 데이터를 보낼 때 걸리는 지연 시간을 측정하였다. WWW 서버와 제어 서버의 업데이트 시간(update time)은 3초이다.

Fig. 4.7은 클라이언트가 WWW 서버에 접속하는 화면이다.

Fig. 4.8은 WWW 서버의 데이터 전송화면이고, Fig. 4.9는 제어 서버의 데이터 전송화면이다.

각각의 지연 시간은 Fig.4 .10에서 나타내고 있다.

WWW 서버에서 제어 서버간의 평균데이터 지연 시간은 1.93초이고, 제어 서버와 WWW 서버간의 평균데이터 지연 시간은 1.88초이다.

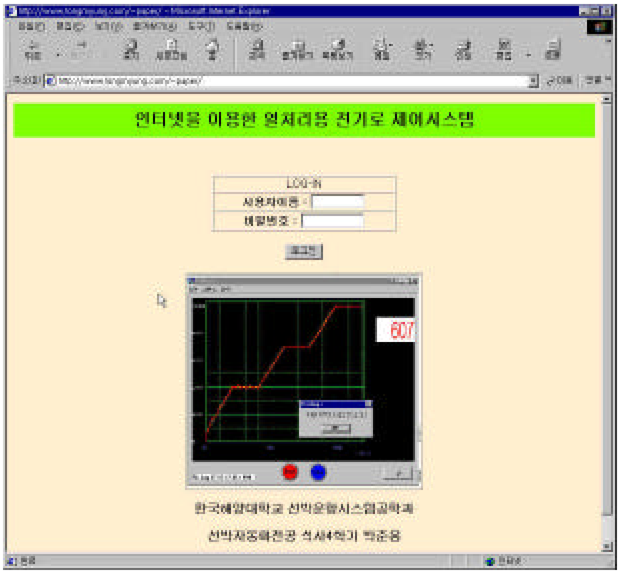

Fig. 4.7 Log-in window of WWW server

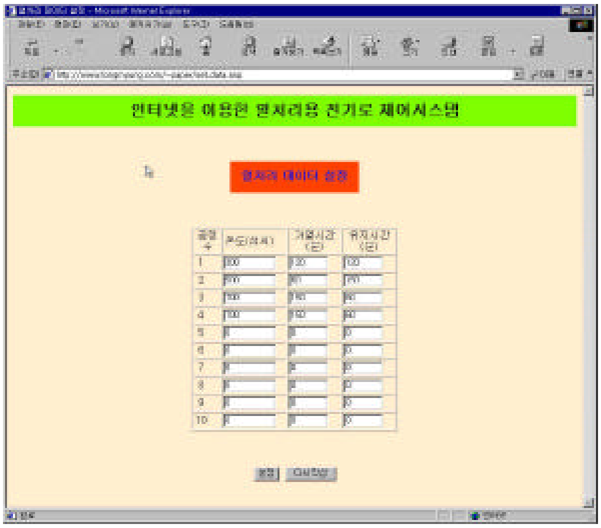

Fig. 4.8 Data transmission window of WWW server

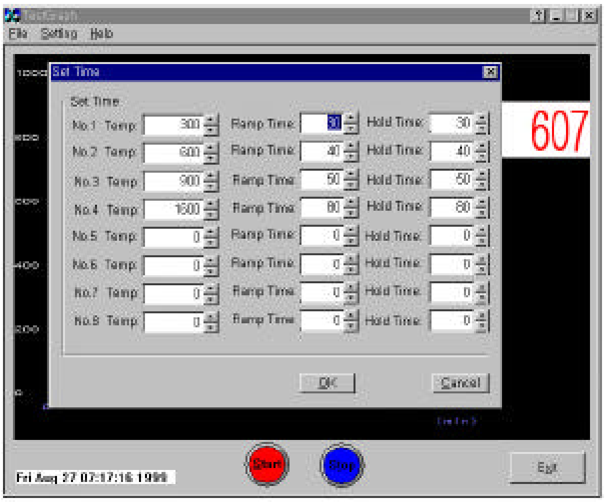

Fig. 4.9 Data transmission window of control server

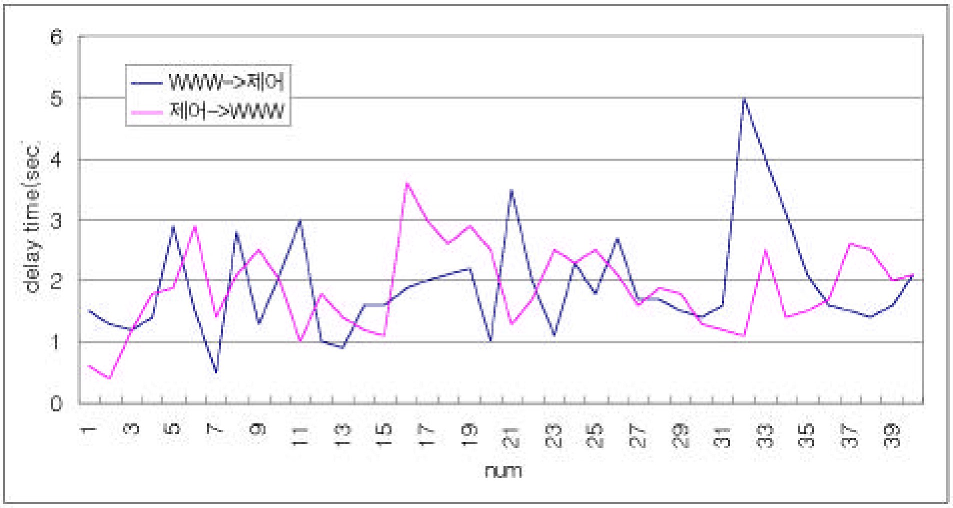

Fig. 4.10 Network delay time

4.3 전기로 특성 실험

이번에 실험은 전기로의 특성을 알아보는 실험으로써 2장에서 설계된 자동동조기법을 이용한 PID 제어기의 파라메타를 실험적으로 구한 다음 그 데이터를 이용하여서 직접 전기로를 제어한다.

제어 서버와 제어기를 시간 지연이 없는 전용망으로 연결하고 전기로를 구동시켰다. 전기로 구동 온도대역은 잠열에 의해서 오차가 많이 발생하는 저온대역과 응답속도가 느리고 온도 변화폭이 적은 고온대역으로 구분하여서 2개의 공정을 동시에 실험을 한다. 자동동조기법을 이용하여서 반복적인 실험을 통해 얻은 PID 제어기 파라메타 값은 식(2.7)에 대입하여서 구하면 다음과 같다.

$K_{pF} = 9.76$, $T_{iF} = 0.164$, $T_{dF} = 0.014$

샘플링시간 T_s는 5초이고 K_F값은 10이다.

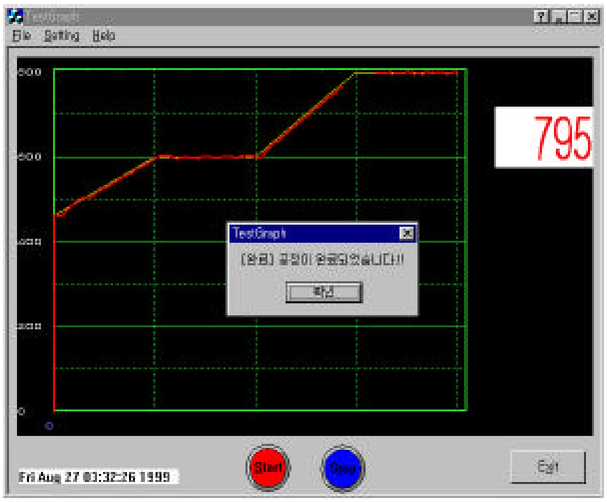

Fig. 4.11 Test result of PID controller

위의 파라메타를 이용하여서 전기로를 제어한 결과는 Fig.4.11과 같다. 저온대역(600℃이하)에서는 ±3℃이하의 오차가 발생했고, 저온대역 이외에서는 ±1.5℃이하의 오차가 발생했다. 그러므로 고정밀, 고신뢰도의 열처리용 전기로 제어장치에 적용하면 좋은 성능을 발휘할 수 있을 것이다.

4.4 제어기 성능 실험

본 제어기 성능 실험은 4.2절 및 4.3절에서 수행한 특성실험 결과를 기초로 하여 인터넷을 이용한 열처리용 전기로 제어기 성능실험을 수행하였다.

클라이언트가 WWW 서버에 접속하여 열처리 데이터를 설정하고 그 데이터를 이용하여 전기로 제어 시스템을 동작 시켰다. 그런 후 제어 서버에서는 전용망 포트를 통해서 제어기에 데이터를 전송했다.

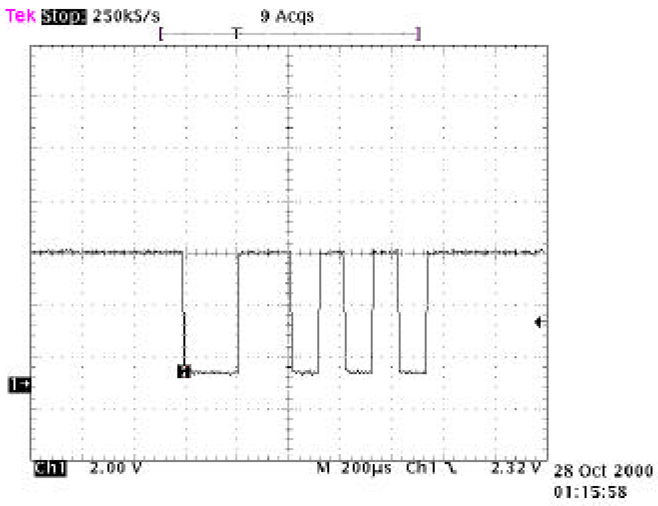

Fig. 4.12 Output voltage waveforms of data transmission

Fig. 4.12는 전용망 포트에서 데이터를 전송하는 신호파형을 측정한 것이다. 그리고 Fig. 4.13은 제어기의 스위칭 모듈에서 전기로를 동작시키는 신호를 측정한 것이다.

인터넷을 이용한 열처리용 제어기 시스템을 이용하여서 전기로를 제어하였다. 네트웍 데이터 지연 시간 T_D는 식(4.1)을 만족하였고 전용망에서는 시간 지연이 전혀 발생하지 않았다. 온도 오차 또한 저온대역에서는 ±3℃이하의 오차가 발생했고, 저온대역 이외에서는 ±1.5℃이하의 오차가 발생했다. Fig.4.14는 인터넷을 이용하여서 전기로를 제어한 그림이다.

$$T_u - T_D > 0 \qquad (4.1)$$

식(4.1)에서 T_u은 제어 서버 업데이트 시간이다.

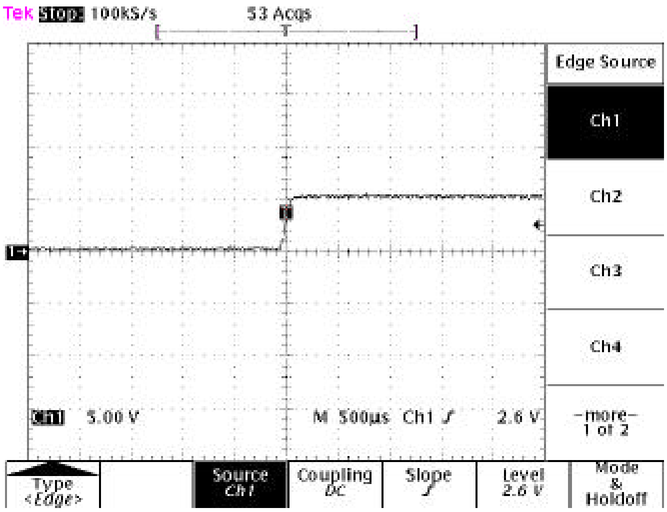

Fig. 4.13 Output voltage waveforms
of switching signal

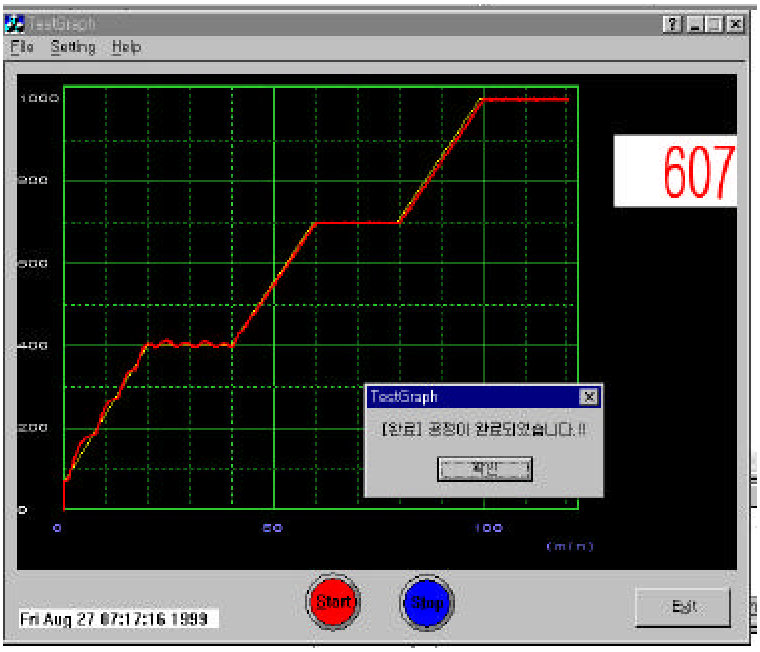

Fig. 4.14 Test result of control using internet

제5장 결론

본 논문에서 구현한 인터넷을 이용한 열처리용 전기로 제어 시스템은 기존의 열처리용 전기로 시스템의 기능을 인터넷의 범위까지 확장한 시스템으로 인터넷이 연결된 곳이면 웹브라우저를 사용하는 것이다. 본 시스템에서는 인터넷망에 특별히 전기로 전용제어기를 설계, 제작하여 제어기용 전용망을 설계하여 고정밀, 고신뢰도의 전기로 제어를 수행할 수 있도록 하는 것이다. 실험을 통하여서 다음과 같은 결론을 얻었다.

1. WWW 서버, DB 서버 및 제어 서버를 구축하고, 인터넷을 이용한 열처리용 전기로 제어한 결과, 원하는 제어성능을 얻을 수 있었다.

2. 인터넷 제어 장치에 전용망을 추가하여 전용제어기를 설계하면 네트웍지연 시간으로 인한 전기로 제어오차를 줄일 수 있음을 실험을 통하여 확인하였다.

3. 열처리용 전기로 제어기 알고리즘에 자동동조기법을 사용하면 열처리 제어기 설계가 용이하다는 것을 실험을 통하여 확인하였다.

본 연구를 통하여 선진국에서 상용화되고 있는 인터넷 제어 기술을 국산화가 가능하게 되었다.

향후 과제로는, 인터넷 제어 기술을 실제 산업분야에 적용시켜 봄으로써

신뢰성 실험을 보강하고 크래킹(cracking)**에 대한 피해를 막기 위해 보안 시스템 구축을 과제로 남긴다**.

참고문헌

(1)이상엽, “Internet programing Bible ", 영진출판사, 1998.

(2)이하섭, "Distance Exhibition System using the telerobotics Technique On World Wide Web", 한국과학기술원, 석사학위논문, 1996.

(3)Wright Stevens, "TCP/IP Illustrated , vol 2 ", Addison Wesley Publishing company,2000.

(4)박영주, “Implementation of the Internet-based Real Time Automated Vehicle Location System", 광주과학기술원, 석사학위논문, 1998.

(5)윤희상, “The Design and Implementation of a Remotely-sensed Image Processing System Using Internet", 한국과학기술원, 석사학위논문, 1997.

(6)김동욱, The Design and Implementation of Scheme for Moritoring the Process Computer, 포항공과대학교, 석사학위논문, 1996.

(7)윤종호, “고속 이더넷”, Ohm사, 1996.

(8)박인규, “A Study on the Development of ships Engine SCADA System using the Internet", 한국해양대학교, 석사학위논문, 1998.

(9)Bong-Kuk Lee, Uk-Youl Huh, "The design of a self tuning controller for the electric furnace temperature control", 대한전기학회 논문지 42권 2호, pp113 121,1993.

(10)S. Omatu, "Adaptive control of furnace temperature by using a micro -computer", IECON'91, pp1979 1984, 1991.

(11)Lyn robison, "Visual C++6 Database Programming", SAMS, 1998.

(12)George Reese, "JDBC and JAVA", O'REILLY, 1996.

(13)Steve Oualling, "Practical C++ Programming", O'REILLY, 2000.

(14)**이상엽**, “Windows Programing bible", **영진출판사**, 1999.

(15)**전병선**, “Microsft Visual C++6.0 MFC Programming", **삼양출판사**, 1999.

(16)**사이버게이트**8, “ **한글윈도우** NT**에서의** ASP**활용**”, **사이버 출판사**, 1999.

(17)John Lewis, William Loftus, "Java software solution" Addison wesley Publishing company, 2000.

(18)Willis J.Tompkins, John G.Webster, "Interfacing sensors To the IBM PC", Prentice hall Inc., 1998.

(19)**컴파일 테크놀로지**, “PIC16C7X **테크니컬 핸드북**”, Comfile Technology, 1998.

(20)CCS Inc., "PCB, PCM and PCW PIC C Compiler Reference Manual", CCS Inc., 1998.

(21)Richard S.Figliola, Donald E. beasley, "Theory and Design For Mechanical Measurments", John wiley&sons Inc., 1991.

감사의 말씀

본 논문이 완성되기까지 2년동안 정성을 다하여 지도해 주시고 아버지처럼 시종일관 용기와 격려로 보살펴 주신 오진석교수님께 진심으로 감사드립니다. 아울러 논문심사를 위해 바쁘신 중에도 기탄 없는 지도와 조언을 주시고 상세한 검토로 미비한 곳을 다듬어 주신 노창주교수님, 이상태교수님께 깊은 감사를 드립니다.

또한 학위과정 동안 무언의 격려와 함께 물심양면으로 도와주신 동명중공업서비스(주) 이상재 사장님께도 감사를 드립니다.

동기로서 많은 실험을 도와주고 실험실 살림을 잘 꾸려나가는 태환, 밤을 지새며 같이 실험한 학선, 선배로서 묵묵히 옆에서 도와준 철웅이형, 아무런 불평 없이 나를 따라주는 실험실 막내 용주 그리고 현정, 희경에게 감사의 뜻을 전합니다.

그리고 대학원 생활동안 많은 도움을 준 큰매형, 큰누나에게 고맙다는 말씀을 드리고 작은매형, 작은누나, 동생 마지막으로 남다른 애정으로 길러주신 어머니와 가장 존경하는 아버지에게 감사의 말씀을 드립니다.

2000년 12월

박 준 용 드림