

## 발전소 보일러 제어용 진보된 제어 알고리즘의 구현

김성우 서창준 김병국  
한국과학기술원 전기및 전자공학과

## Implementation of Advanced Control Algorithms for a Power Plant Boiler System

S.W. Kim C.J. Seo B.K. Kim  
Dept. of Electrical Engineering, KAIST

## Abstract

This paper describes a duplexed process control system for a boiler-turbine system of power plant, which is composed of supervisor, controller, and simulator subsystems. Its embedded POL (problem oriented language) as a structured control language enables this system to do a real-time distributed control and fault diagnosis by simple programming with the actual implementation of advanced control algorithms such as PID autotuning and GPC, etc, the performance of overall system has been greatly enhanced.

## 1 서론

산업 사회의 발달과 더불어 전력 산업도 매우 복잡화하게 되었고, 그 파급효과도 커지게 되었다. 특히, 발전소 제어 시스템의 안정성 있는 운용이 중요한 문제로 부각되고 있다. 그래서, 여기에 발전된 현대 제어 이론을 적용하려는 노력으로 기존의 많은 아날로그 제어 시스템이 디지털 분산 제어 시스템으로 대체되고 있다.

현대적인 디지털 분산 제어 시스템은 보일러 시스템 같은 대규모 공정 시스템에 적합하다. 이는 전체 프로그램을 모듈화하여 주기적으로 매 샘플링 시간 동안 공정제어를 수행함과 동시에, 고장 진단 및 기타 작업이 분산 병행하여 이루어질 수 있기 때문이다. 이미 개발된 디지털 분산 제어 시스템은 위의 기능을 거의 모두 갖추면서, 아울러 고신뢰도를 이룩하기 위해 redundant system을 채용한 경우가 Bailey 등의 제품에서 찾아볼 수 있다[2, 3]. 본 연구에서는 이같은 디지털 분산 제어 시스템에서 중요한 역할을 담당하는 공정 제어 시스템을 구현하

였다[1]. 이 시스템은 실시간 제어를 위하여 실시간 운영체제인 VxWorks를 사용하고 있고, 이중화된 프로세서는 실시간 고장 진단 task와 결합되어 고장 진단 및 대처 기능을 강화하여 고신뢰성을 갖게 하였다.

또한, 대부분의 multi-loop 공정 제어기에 적합한 기능 블록(function block) 개념의 모듈화된 제어 언어를 구현하였다[1]. 여기서, 기능 블록이란 개발자가 구현한 어떤 기능을 갖는 호출 가능한 함수를 말한다. 이러한 시스템 구조를 갖추고 난 뒤 사용자는 기능 코드로 구성된 공정 제어 프로그램을 짜기만 하면 된다.

발전소 제어용 공정 제어 시스템이 구현되고 나면, 실제 보일러 시스템에 적용할 수 있는 진보된 제어 알고리즘이 필요하게 된다. 진보된 제어 알고리즘에는 여러가지가 있겠으나, 우선 많이 쓰이는 PID제어기의 자동 동조와 모델 예측 제어를 고려할 수 있다[7, 8]. 본 논문에서는 MIMO보일러 시스템에 적합한 PID 제어기의 자동 동조 알고리즘과 일반형 예측 제어를 기능 코드로 구현하였다. 뒤에서 이러한 제어기들을 직접 시뮬레이션함으로써 실제 적용 가능성을 확인하였다.

## 2 이중화된 공정 제어 시스템

시스템의 호스트 컴퓨터는 OIU/EWS (Operator Interface Unit/ Engineering WorkStation)기능을 갖는 SUN 워크스테이션을 채택하였고, 프로세서 모듈로는 모듈 자체에 Ethernet Port를 내장하고, 68030 CPU에 의한 고속의 연산 능력을 지닌 Force사의 CPU30을 채택하였다.

CPU30 보드는 호환성 및 사용도가 높은 VME BUS에 인터페이스되어 이더넷을 통해 SUN 호스트 컴퓨터와 정보를 교환한다[4]. 이중화를 위해 주 프로세서 모듈 외에 백업 프로세서 모듈이 있으며, 발전소 보일러 시스템을 이산 시스템으로 구현한

시뮬레이터 모듈이 있다. 결국, 이 시스템은 3개의 CPU30 보드로 구성되게 된다.

호스트 컴퓨터는 기본적인 UNIX 운영 체제 하에서 모듈화와 Multi-tasking이 가능한 실시간 운영 체제 (Real-time OS)인 VxWorks를 도입하여 대규모 공정 제어 시스템을 구성하였다. VxWorks는 호스트 컴퓨터에 올려져 있으며, 타겟 프로세서인 CPU30이 부팅되면, VxWorks관련 파일들이 LAN을 통해 RAM으로 올려져 실시간 제어에 이용된다. 발전소 보일러 제어용 프로그램을 프로세서 모듈에서 실행한 결과를 X Window상에서 나타냄으로써 operator interface가 가능하게 되었다. 전체적인 하드웨어 시스템 구조는 그림 1과 같다.

본 연구에서 구현된 발전소 보일러 디지털 공정 제어 시스템의 프로세서 모듈은 이중화되어 신뢰도를 높였다. 이중화된 프로세서 모듈의 고장 진단, 대처 기능에 대한 연구는 앞으로 계속되어야 할 과제이다.

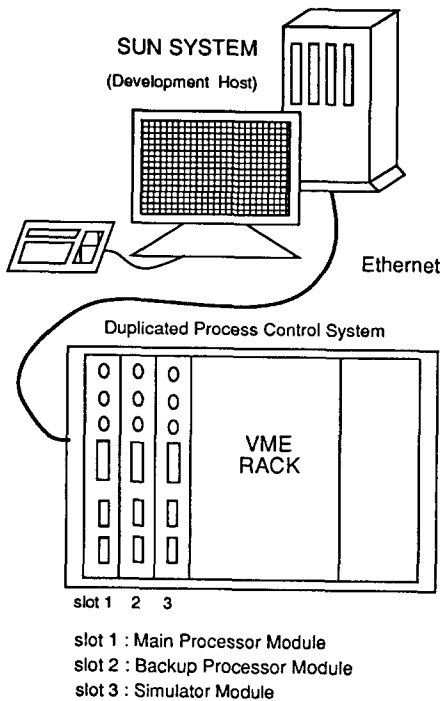


그림 1. 전체 시스템 구성도

### 3 제어 언어 생성기

대부분의 공정 제어 형태는 여러 개의 local loop의 복합적 구성으로 이루어진다. 경제적인 측면에서 다수의 single-loop 제어기보다는 이러한 복합 loop을 일괄 취급하는 제어기가 필요하게 되는데, 이것이 multi-loop 제어기이다.

Multi-loop 제어기는 compact geometry, computing power, 복합적 control strategy, 그리고 좋은 성능 등이 요구된다. 제어 기능으로는 여러개의 PID 제어 기능 이외에 adaptive filtering, lead/lag compensation, signal processing 등의 다양한 기능을 내장해야 한다. 또한, 대규모 공정 제어를 위해 기능의 모듈화에 의해 확장성을 갖고, 복잡한 제어 시스템의 구성이 가능하게 하여야 한다[1, 2].

Configurable controller는 function block 개념의 언어로 프로그램(configuration) 하는데, 이 function block들은 control block diagram에서 각각의 block에 해당하는 기능들로서, 자주 사용되고 유용한 기능들을 개발자가 function subroutine 들로 미리 구현해놓고, 사용자는 이들을 유기적으로 결합할 수 있게 하는 일종의 POL(Problem Oriented Language)이다. 사용자는 각 function block내의 구조를 알 필요가 없으며, 단지 그 입력, 출력, 기능만을 이해하면 된다.

기능코드로 된 configuration 파일을 타겟 프로세서가 직접 이해할 수 없으므로 이식성이 강한 C 언어 프로그램으로 변환시킨 다음 타겟 프로세서에 맞는 cross compiler를 써서 기계어로 바꾸어 주게 된다.

configuration 파일로부터 C 언어 프로그램으로의 변환은 UNIX 제공 유틸리티인 lex (lexical analyzer, 어휘 분석기)와 yacc(syntax/semantic analyzer, 어의 해석기)를 이용하여 구현된다. 생성 방법은 각 기능 코드에 해당하는 함수를 C 언어로 작성하고, configuration 파일의 기능코드 순서에 따라 함수들을 호출하는 함수를 작성하는 것이다. C 언어 프로그램 생성의 기본적인 구조는 다음 그림 2와 같다.

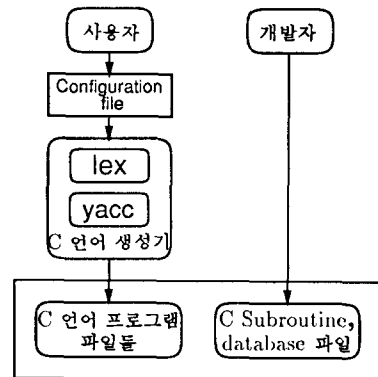


그림 2. C 언어 프로그램 생성의 전체 흐름도

function block들은 기능 코드(function code)들로 표현 되는데, 여기서 기능 코드는 function block들의 입출력 관계와 파라미터들을 사용자가 알기 쉽게 ASCII 문자로 표현한 것

이다. 제어기의 구성은 기능코드들의 나열로 이루어진 configuration 파일을 작성함으로써 이루어진다. 현재까지 구현된 발전소 보일러 제어용 기능 코드들은 80여개 정도이다. 기능 코드의 형식은 다음과 같다.

블럭 번호 = 기능코드 이름(입력포트 번호들;파라미터 값들)

간단한 PI 제어기를 그림으로 나타내면 다음과 같다.

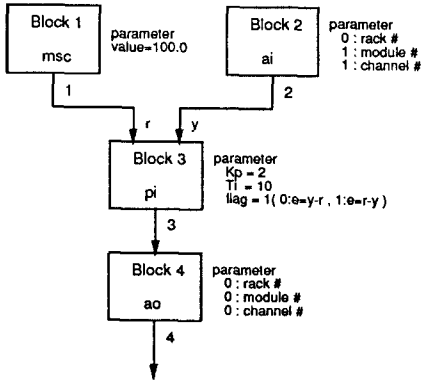


그림 3. 간단한 PI제어기에 대한 block diagram

위 예제 그림의 Configuration file을 아래에 나타내었다.

```
! This is a example of configuration file
#define MAX_BLK_NUM 4
#define MAX_OUT_NUM 4

1 = msc(100.0)           ! manual set constant
2 = ai(0,1,1)           ! analog input
3 = pi(1,2,2,10,1)     ! PI controller
4 = ao(3;0,0,0)        ! analog output

! End of example file
```

그림 4. 간단한 PI제어기에 대한 configuration file

구성된 제어기에 의해 플랜트가 제어되고 있을 때, 보다 좋은 플랜트의 응답을 위해서 온라인(On-line) 파라미터 튜닝이 가능하게 되어 있다. 파라미터 튜닝 과정은 다음과 같다. 우선 제어기 초기화 루틴에서 파라미터 형에 대한 정보를 저장한 후 제어기가 실행되어 튜닝정보가 타겟 프로세서로 다운로드되면 미리 만든 tuning task는 교체될 파라미터 memory 위치에 새로운 파라미터 값을 대체함으로써 튜닝이 이루어진다.

여기서, 파라미터가 있는 모든 기능코드에 대해 튜닝이 가능하며, 튜닝 정보는 튜닝될 기능 코드 블럭 번호, 파라미터 갯수, 값, 순서, 데이터 전달 확인을 위한 checksum 그리고, 튜닝 정보의 유효flag 등이다.

## 4 진보된 제어 알고리즘

### 4.1 PID 자동 동조

오늘날 많은 고급 제어 이론들이 발전되고 있음에도 불구하고, PID제어가 여전히 많이 쓰이고 있다. 이는 PID제어기가 연산이 간단하고 강인성 등의 장점을 갖기 때문이라고 볼 수 있다.

여기서 PID계수들은 제어 대상 프로세스의 정보에 의존하기 때문에 동조하는 작업에 많은 노력이 필요하다. 다행히도, 프로세스 정보가 별로 없이도 PID계수를 자동으로 동조하는 알고리즘이 많이 제안된 바 있다[7]. 본 연구에서는 Relay feedback의 describing function을 이용한 자동 동조 방법용 기능 코드로 구현하였다.

그 구성도는 다음과 같다.

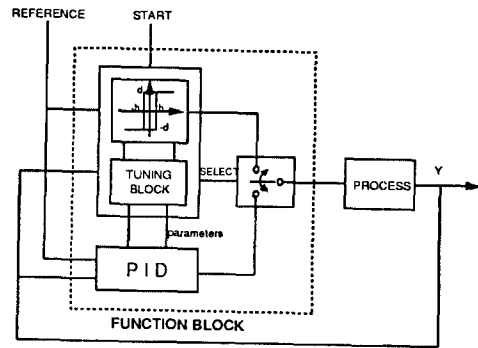


그림 5. PID 자동 동조

### 4.2 일반형 예측 제어

미래의 어떤 구간 내에서의 공정 출력을 예측함으로써 원하는 공정 출력 오차와 제어입력을 최소화하는 이동 구간 제어 방식을 일반형 예측 제어 (GPC, Generalized Predictive Control)라 한다. 일반형 예측 제어는 선형화된 모델 파라미터가 충분히 인식되었다는 가정 하에 비최소위상 공정, 불안정한 공정, 시간 지연을 갖는 공정 등에 모두 적합한 보편적인 제어 방식으로 알려져 있다[8]. GPC는 performance-oriented MIMO adaptive GPC 등 변형된 많은 알고리즘이 개발되어 있으며, 앞으로도 더욱 심화된 연구가 있을 것으로 사료된다[9].

보일러 시스템에 GPC를 적용하기 위하여 크게 두 가지 제어 방식을 생각할 수 있다. 첫째는, 근사화된 전체 보일러의 동특성에 맞는 MIMO GPC를 적용하는 것이고, 둘째는, 압력, 온도, 수위, 노내압 등의 여러 동특성으로 구성된 보일러 시스템

에 multi loop GPC 제어를 적용하는 것이다. 각각의 동특성의 선형성이 크고, 기존의 multiloop PID 제어를 GPC로 대체할 수 있다는 점 등에서, 전자보다는 후자가 더욱 적용 타당성이 크다. 그러나, 본 연구에서는 전자만을 고려해 MIMO adaptive GPC를 기능코드로 구현하여 시뮬레이션하였다. 후자의 경우는 차후 연구되어질 것이다. 시뮬레이션에서 사용된 보일러 모델은 Bell과 Åström의 160MW 화력 발전소 드럼형 보일러 모델이다[5,6]. 모델식은 다음 (1),(2),(3),(4)식이다.

$$\begin{aligned} \dot{x}_1 &= -0.0018u_2x_1^{\frac{5}{3}} + 0.9u_1 - 0.15u_3 \\ \dot{x}_2 &= (0.073u_2 - 0.016)x_1^{\frac{5}{3}} - 0.1x_2 \end{aligned} \quad (1)$$

$$\dot{x}_3 = (141u_3 - w_s)/85$$

$$w_s = (1.1u_2 - 0.19)x_1$$

$$y_1 = x_1, \quad y_2 = x_2$$

$$y_3 = 0.05(0.13073x_3 + 100a_{cs} + q_e/9 - 67.975)$$

$$a_{cs} = \frac{(1 - 0.001538x_3)(0.8x_1 - 25.6)}{x_3(1.0394 - 0.0012304x_1)}$$

$$q_e = (0.854u_2 - 0.147)x_1 + 45.59u_1 - 2.514u_3 - 2.096$$

$$0 < u_1 \leq 1, \quad |\dot{u}_1| \leq 0.007/sec \quad (2)$$

$$0 < u_2 \leq 1, \quad -2/sec \leq \dot{u}_2 \leq 0.2/sec \quad (3)$$

$$0 < u_3 \leq 1, \quad |\dot{u}_3| \leq 0.05/sec \quad (4)$$

위에서, 제어 입력  $u_1, u_2, u_3$ 는 각각 보일러 연료 조절기 밸브, 터빈으로 가는 증기량 조절 밸브 그리고 급수 조절기 밸브 위치를 나타내며, 위 (2),(3),(4)식과 같은 제한식을 갖는다. 또,  $y_1, y_2, x_3, y_3$ 은 각각 드럼 증기압 ( $kg/cm^2$ ), 전기적 출력 ( $MW$ ), 드럼 내의 유체의 농도( $kg/cm^3$ ), 드럼의 수위( $m$ )를 나타낸다.

## 5 시뮬레이션 및 결과 검토

### 5.1 PID 자동 동조

Simulation을 위해 사용된 플랜트는 다음과 같다.

$$\frac{Y(s)}{U(s)} = \frac{0.03}{(s + 0.3)(s + 0.1)}$$

제어 입력은 -5와 5사이에서 동작하도록 제한하였으므로, PID 제어기는 이 구동기 포화로 인한 anti-reset windup 현상을 보상하고, 미분항에서 고주파 성분을 제거할 수 있어야 한다. 더불어, 비례항에서 기준 입력과 출력의 비율 다르게 하여 강인한 특성을 가질 수 있게 되었다.

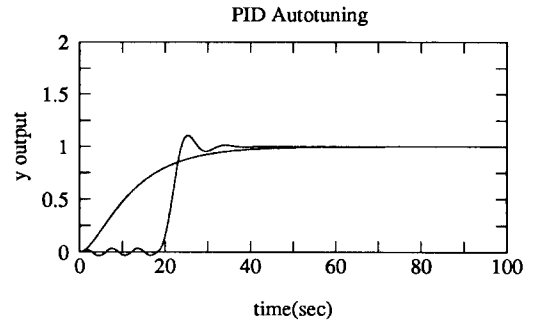


그림 6. PID 자동 동조에 대한 시뮬레이션 결과

그림 6의 시뮬레이션 결과는 화면 상의 결과를 그대로 출력한 것이다. 기능 코드로 구현된 자동 동조 알고리즘은 아무 이상 없이 잘 동작한다. 여기서 보듯이, 제대로 동조된 PID 제어기는 개루프 특성에 비해 훨씬 좋은 응답을 보이고 있다. 동조 후 PID 계수는  $Kp = 21.95, Ti = 3, Td = 0.75$  이다.

### 5.2 일반형 예측 제어

앞에서 서술한 (1)식의 비선형 보일러 플랜트에 MIMO adaptive GPC를 적용하였다.

$\hat{A}(q^{-1}), \hat{B}(q^{-1})$ 에서 추정 최대 차수는 2이며, 제어되는 출력 변수는  $x_1, x_2, x_3$ 이다. 파라미터 추정은 확장된 RLS 알고리즘을 썼고, 출력 예측 범위  $N = 20$ , 제어 범위  $Nu = 1$ , 그리고  $\lambda = 0.1$ 로 하였다.

비선형 플랜트로 인한 초기 파라미터 추정의 불안정성 문제 해결을 위해 미리 학습된 초기 동작점에서의 선형 모델 파라미터를 설정해 주었다. 그리고, GPC의 선형 모델로 인한 unmodeled dynamics를 피하기 위하여 급격한 계단 입력 대신 완만한 램프 입력 형태의 기준 입력을 주었다. 기준 입력은 처음 200초 동안 100%의 공칭 동작점에서 머물러 있다가, 서서히 증가하여 110%에 머물렀다가, 다시 100%로 감소하는 것이다. 시뮬레이션 결과는 그림 7과 같다.

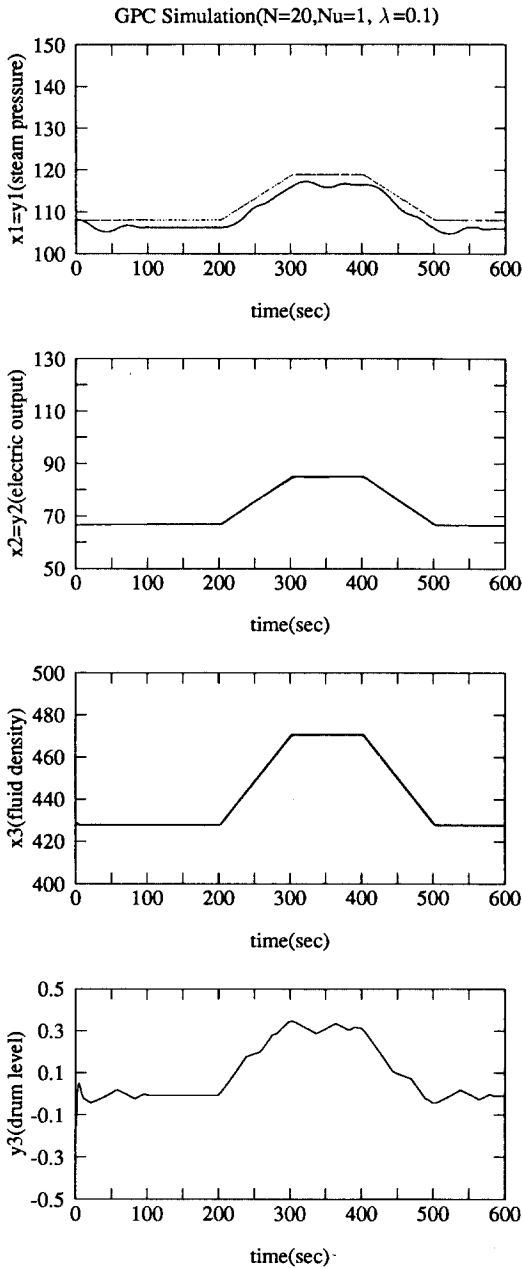


그림 7. GPC 시뮬레이션 결과

그림 7의 결과를 보면, 동작점의 변화에 따른 출력의 추종 성능이 비교적 좋은 것을 볼 수 있다.  $y_1$ 의 정상 상태 오차가 생기는 것은 GPC의 출력 예측 범위가 큰 데 기인한 계산 오차 때문이다. 결국, 이 비선형 보일러 모델은 출력 예측 범위에 민감한 특성을 갖는다.

## 6 결론

본 논문에서는 이중화된 PCS와 제어 언어 생성기 및 고급 제어 알고리즘의 구현에 대해 서술하였다. 이로써, 고장 진단, 대처 기능이 강화된 이중화 디지털 분산 제어 시스템이 구축되었으며, 실시간 운영 체제를 이용한 발전소 보일러의 고급 제어가 가능해졌다. 최종적으로 고장 진단용 소프트웨어와 고급 제어 프로그램 모듈의 결합 및 보일러 시스템 시뮬레이터의 보완을 통하여 실제 보일러 시스템에 효과적으로 적용하는 문제가 남아 있다.

산업체에서 많이 쓰이는 PID 제어기의 자동 동조와 GPC 제어의 구현은 많은 다른 진보된 제어 알고리즘의 실제 적용 가능성을 시사하고 있다. 앞으로, 보다 진보된 제어 알고리즘에 대한 이론적인 연구가 계속 이루어져야 한다.

## 참고문헌

- [1] 한국 전력 공사 기술 연구원, "분산 제어 시스템의 고장 대처 기능 및 제어언어의 구현(중간 보고서)", 1992. 2.
- [2] Bailey Controls, "Bailey Network 90 - Function Code Reference Manual".
- [3] Bailey Controls, "C Language Implementation Guide for the Multi-Function Controller (NMFC03) - Software Manual".
- [4] Force Computers Inc., "SYS68K/CPU-30 User's Manual", Revision No.3, 1989
- [5] 권 옥현, 김 상우, 박 부균, 김 은기, "개선된 LQG/LTR 방법에 의한 보일러-터빈 제어 시스템의 설계", 전기 학회 논문지 39권 2호, 1990
- [6] R.B. Bell and K. J. Åström, "Dynamic models for boiler-turbine-alternators units: data logs and parameter estimation for a 160MW unit," Report TERT-3192, Lund Institute of Technology, Sweden, 1987
- [7] K. J. Åström and T. Hägglund, "Automatic Tuning of PID Controllers", ISA, 1988
- [8] D. W. Clarke, C. Mohtadi and P. S. Tuffs, "Generalized Predictive Control-Part I. The Basic Algorithm", *Automatica*, vol. 23, No. 2, pp 137-148, 1987

- [9] M. Kinnaert, "Adaptive generalized predictive controller for MIMO systems", *Int. J. Control*, vol. 50, No. 1, 161-172, 1989