

DCS에 퍼지제어 알고리즘 구현 방법에 관한 연구

°희 윤기*, 변 중남**

*포항종합제철 기술연구소, **KAIST 전기 및 전자공학과

A Study on Realization method of Fuzzy Control Algorithm for DCS

°Yone Gi Hur*, Zeungnam Bien**

*POSCO Technical Research Laboratories, **KAIST Dept. of Electrical Engineering

ABSTRACT

As the modern industrial processes become more complex, it is getting more difficult to model and control the processes. Naturally, an advanced type of DCS(Distributed Control System) with higher level functions is being sought. Advanced DCS is a DCS with advanced functions such as fault diagnosis, GPC(Generalized Predictive Control), NN(Neural Network), and Fuzzy Control.

In this thesis, we have studied a fuzzy control algorithm for realizing an advanced DCS. Its algorithm is implemented in a form of function code which is a process control language, being used by the industrial engineers.

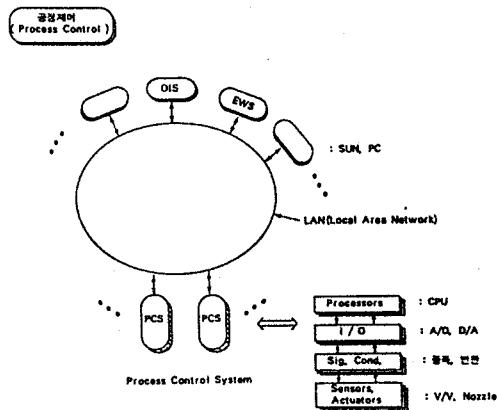
To verify the realized function code of the fuzzy control, the function code is applied to a continuous casting process of the Pohang Iron & Steel Works in Kwangyang. The rules of the fuzzy control were collected via interviews of the field operators and their operation documents.

Finally, usability of the function code of the fuzzy control is shown via simulation for the continuous casting process model.

1. 서 론

발전소와 제철소 등과 같은 대규모 공정제어분야에 디지털 제장제어 시스템이 도입되어 있으며, 과거 아날로그 제어방식에 비해 분산제어시스템(DCS, Distributed Control System)은 제어 시스템의 안정성 있고 효율적인 운용을 위해서 필요하며 그의 중요성은 강조하지 않아도 이미 산업체에서 많이 인식되어 있다. DCS는 (그림 1)에 개념을 간략화하여 도시하였으며 간단히 설명하면 다음과 같다. DCS는 OIS(Operator Interface Station)와 EWS(Engineering Work Station)과 PCS(Process Control System)의 3부분으로 구성되어 있다. OIS는 오퍼레이터와 공정이 운용되고 있는 하드웨어(PCS)간의 인터페이스를 담당한다. 오퍼레이터는 공정에서 운용되는 설정치(SV, Set Value) 및 제어

입력(MV, Manipulated Value) 그리고 공정출력값(PV, Process Value) 등을 모니터링 한다. 또한 알람신호 및 계이터의 파라미터값을 설정하기도 한다. EWS는 엔지니어가 제어알고리즘을 구현하는 곳이다. 본 연구에서 구현하고자 하는 제어알고리즘은 이 곳 EWS에서 이루어진다.



본 연구에서는 DCS에 퍼지제어 알고리즘을 구현하는 방법과 이의 검증작업으로 광양제철소 1연주공장을 대상으로 하여 공정을 모델링하고 실시간 모의실험한 결과를 소개하고자 한다.

본 논문의 구조는 다음과 같다. 제1장의 서론에 이어서, 제2장에서는 DCS에 퍼지제어 알고리즘을 구현하는 방법 및 기능코드에 대해서 서술한다. 제3장에서는 제어대상인 제철소 연주공정의 모델링과 주조시작과 종료시의 퍼지제어기 구성과 제어규칙을 소개한다. 제4장에서는 퍼지제어 기능코드의 검증을 위하여 연주공정을 대상으로 실시간 모의실험한 결과를 소개한다. 제5장에서는 연구결과를 기술한다.

2. 퍼지제어를 위한 기능코드

제어 알고리즘을 DCS에 구현하려면 공정제어언어인 기능코드의 형태가 되는 것이 효율적이다. 대규모 공정의 제어를 위

여 제어기 구성에 있어서 모듈(Module)화에 의해 확장성을 갖고 복잡한 제어 시스템의 구성을 용이하게 하는 것이 필요하다. 이러한 개념으로 구성된 것이 컨피그레이션 파일(configuration file)이다. 이는 기능블록(Function block)개념의 언어로 구성되는데, 이 기능블록에 해당하는 기능은 개발자가 기능 서브루틴(Function subroutine)을 미리 구현해 놓게 된다. 사용자는 이 기능블록을 유기적으로 결합하여 제어알고리즘을 구성한다. 이때 사용자는 기능블록의 입력, 출력, 기능을 이해하면 된다. 기능블록은 기능코드(function code)로 표현된다. 제어알고리즘을 구현하려면 컴퓨터언어로 코딩해야하는 테, 이 때 기능 별로 서브루틴(Subroutine)을 작성해놓으면 사용자는 자신이 구현하고자 하는 제어알고리즘에 맞게 그의 서브루틴을 사용하면 되는 것이다. 이렇게 공정에 필요한 기능에 따라서 작성해놓은 서브루틴을 기능코드라고 한다. 기능코드를 해야하는 이유는 사용자가 편리하게 제어알고리즘을 구현하고 제어알고리즘 구현의 일반성을 부여하기 위해서이다. 기능코드는 기능블록이 갖는 블록들 사이의 입출력 관계와 내재된 파라메터(Parameter)들을 사용자가 알기 쉽게 표현한 것이다. 제어기의 구성은 기능코드의 나열로 이루어 진다. 이때 기능코드의 나열로서 이루어진 파일을 컨피그레이션 파일이라 한다.

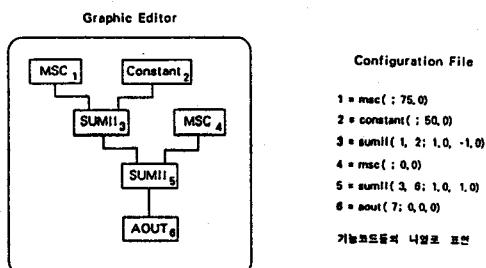


그림 2 그레픽 에디터 및 Configuration file

기능코드를 구현한다는 것은 컨피그레이션 파일의 각각의 기능(function)에 해당하는 서브루틴을 코딩함을 말한다. 본 연구에서는 DCS에 퍼지제어 알고리즘을 구현하는 방법을 제시하고자 한다. 공정의 규모가 커지면 제어 알고리즘이 그만큼 복잡하게 된다. 따라서 기능코드의 형태로 모듈화되어 있지 않으면, 새로운 알고리즘을 추가하기 어렵고 또한 기존의 알고리즘을 수정하기도 쉽지 않다. 기능코드로 구현되기 위해서는 범용성이 되어야 하고 따라서 기능코드간의 코딩 규칙이 있다. 한편 퍼지제어같은 지능제어의 기능코드의 구현은 쉬운 것이 아니다. 기존의 기능코드에 ffc() 함수 하나 첨가했다고 해서 제어가 되는 것 이 아니다. 지능제어 기능코드의 특징은 기존의 기능코드는 스칼라(Scalar)를 입력받아서, 스칼라를 출력하는 테 반하여, 벡터나 행렬을 수행할 수 있는 기능이 주어져야 한다. 예를 들어, 신경망의 경우 가중치(Weight)를 다루므로 벡터의 기능이 주어져야 하고, 퍼지제어는 퍼지한 값을 다루므로 벡터나 행렬을 처리할 수 있는 기능이 주어져야 한다. 또한 벡터나 행렬의 크기(Size)가 변화가능해야 한다. 만약, 이것이 고정되어 있으면 컴퓨터의 메모리 문제도 있고, 일반화된 기능코드 구현을 위해서는 좋지 못하다. 따라서 그의 크기가 유동적으로 변화가능해서 사용자가 마음대로 지정할 수 있어야 한다.

리할 수 있는 기능이 주어져야 한다. 또한 벡터나 행렬의 크기(Size)가 변화가능해야 한다. 만약, 이것이 고정되어 있으면 컴퓨터의 메모리 문제도 있고, 일반화된 기능코드 구현을 위해서는 좋지 못하다. 따라서 그의 크기가 유동적으로 변화가능해서 사용자가 마음대로 지정할 수 있어야 한다.

3. 연주공정의 퍼지제어기 구성

제철소는 크게 재선공정, 제강공정, 그리고 압연공정의 3단계를 거친다. 연주공정은 제강공정과 압연공정을 연결해 주는 공정으로서, 제강공정에서 나온 용선을 압연을 하기 적합한 주판(Slab)을 만드는 작업을 한다.

연속주조공정(連續鑄造工程, Continuous Casting Process)은 1850년 경 전로의 발명자인 Henry Bessemer의 특히로 처음 소개되었고 1960년대에 주판(鑄片, Slab)의 제조에 적용되었다. 이 공정은 용강(溶鋼, Molten steel)이 최초로 형태를 갖는 공정으로서 열간 압연 공정에 들어가기 위한 강재(鋼材)인 주판을 만드는 공정이다. 1955년 일본에서 최초로 연속주조기가 설치된 이래로, 오늘날 제강공정 중 가장 중요한 공정으로 평가받고 있고 또한 널리 사용되고 있다. (그림 3)은 포항제철 광양제철소 제 1연주 공장의 연속주조공정을 나타낸다.

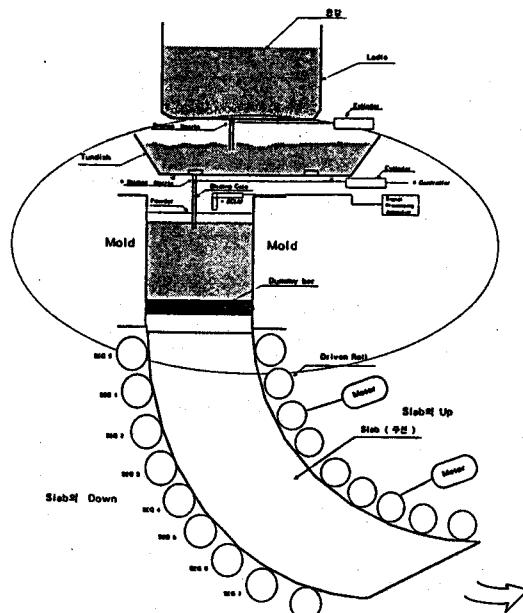


그림 3 제철소 연속주조공정

본 연구는 제철소의 연속주조공정에서의 주조 시작(Start)시 기존의 수동제어에서 자동제어로 향에 그 목적이 있다. 연속주조공정에서 제어의 중요한 부분은 주형(鑄形, Mold)내 용강의 레벨(level)을 제어하는 것이다. 레벨 제어가 제대로 이루어지지 않으면 제어시 큰 오버슈트

(Overshoot)는 주형밖으로 첫물이 분출(overflow)하여 사고의 위험을 초래할 수 있다. 또한 심한 댐핑(Damping)은 주형내 용강의 요동을 야기하여 결국에는 주변의 표면에 손상이 가서 가공하기에 부적합하게 된다. 따라서 제어가 잘되려면 안정되게 용강의 레벨이 원하는 레벨(Reference Level)에 도달해야 한다.

과도상태의 레벨제어는 숙련조업자가 수동제어로 수행하는 데, 퍼지제어는 이러한 숙련조업자의 예매한 지식을 적절하게 이용가능하다. 연속주조공정을 퍼지논리를 이용한 제어로서 자동화하여 운전의 효율을 높일 수 있도록 하는 것이 연구의 목적이다.

주형 레벨제어와 주조속도제어 투프에 관련된 전체적인 블럭도는 (그림 4)과 같다.

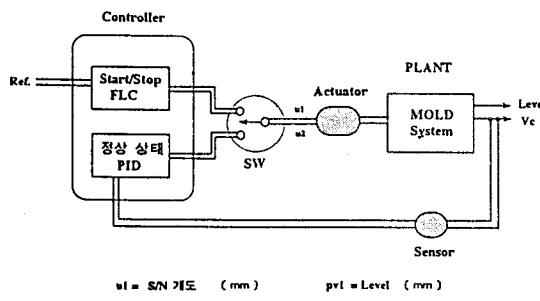


그림 4 주형제어 투프 전체 블럭도

퍼지제어의 제어규칙은 현장에서 조업자들과 대화를 통하여 제어규칙을 얻었다. 아래에는 현장에서 얻은 지식을 규칙으로 표현한 것이다.

● 주형 레벨 제어 규칙

레벨 관련 퍼지제어는 입력이 레벨과 그의 변화량이고 출력은 S/N의 개도이다.

ΔL	L	low	high	Ref_-	Ref_	Over
NB						
NM	PB	PS	PS	PS	ZO	
ZO	PB	ZO	ZO	ZO	NS	
PM	PS	ZO	NS	NS	NS	NB
PB	PS					NB

● 주조 속도제어 규칙

주조 속도제어는 입력이 속도의 오차와 그의 변화량이고 출력은 모터의 전기자전압의 변화량이다.

ΔE	E	NB	NS	ZO	PS	PB
NB				NB		
NS				NS		
ZO	PB	NS	ZO	PS	PB	
PS				NS		
PB				PB		

4. 모의 실험

주형내의 용강의 레벨의 설정치는 800mm이다. 주조속도의 설정치는 현장에서 얻은 데이터를 분석하여 만들었다.

실험환경에 대하여 설명하면 다음과 같다. (그림 5)에서 보이듯이 OIS와 EWS의 기능을 담당하는 SUN이 개발 호스트(Host)이다. 여기서 기능코드와 대상모델이 컴파일(Compile)되어 이서 이더넷(Ethernet)을 통하여 CPU30에 다운로드(download)된다. CPU30은 VME Bus를 통하여 CPU30으로부터 대상모델을 다운로드받는다. CPU30은 퍼지제어 기능코드가 다운로드되어 운영되는 제어부이고, CPU33은 대상모델이 운영되는 시뮬레이터부가 된다. CPU30과 CPU33은 VME Bus를 통하여 데이터를 통신하며 실시간(Real time)으로 운영된다. 여기서 CPU30과 CPU33이 PCS역할을 하게 된다.

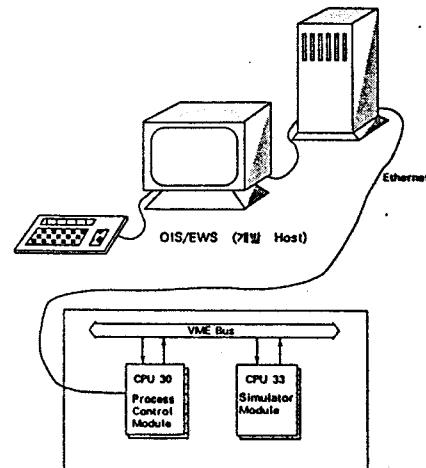


그림 5 실험 환경

대상모델을 제어하기위한 퍼지제어 기능코드가 포함되어 있는 컨피그레이션 파일을 (그림 6)에 소개한다.

Configuration File 소개

```

! configuration file
! PLC Controller

INITIALIZE
1 errV = macf(0.0001)
2 devV = macf(0.0001)
3 uOLC = macf(75.0)
CONTROL
4 L = ainf(0, 0, 0)
5 V = ainf(0, 0, 1)
6 setL = macf(800.0)
7 setub = macf(750.0)
8 setV = refVL_setub()
9 dL = sumif(L, Ldelay, 1.0, -1.0) ! dL = L(tk) - L(tk-1)
10 Ldelay = delay(L, 347.0) ! Ld = L(tk-1)
11 errV = sumif(V, V, 1.0, -1.0) ! errV = e(tk)
12 devV = sumif(errV, errdelay, 1.0, -1.0) ! devV = e(tk) - e(tk-1)
13 errdelay = delay(errV, 0.0) ! errd = e(tk-1)
! PLC of motor velocity
14 devE = fclerv, devV, 5.3, 5.3, 5.5; "RateV.dat"
15 minub = macf(75.0)
16 minU = imax(uVdelay, minub) ! minU = MAX(uV(tk-1), 75.0)
17 uV = sumif(minU, devV, 1.0, 1.0) ! uV = uV(tk-1) + du
18 uVdelay = delay(uV, 75.0) ! uVd = uV(tk-1)
19 nouV = macf(0.0) ! nouV = 0.0

```

```

20 select = mac(750.0)
21 coninV = select(l, lmark, uV, nouV) if coninV = uV, if l > 750.0
22 ouc = mac(75.0)
23 uOLC = ouc(ouc)
1 uOLC = 75.0 .. 0.005
1 FLC of mold Level
24 uFLC = fctL_dL_53_53_53 ; "RuleList"
25 select = mac(730.0)
26 conin = select(l, lmark, uFLC, uOLC) ; l mark .. uFLC, if l > 730.0

1 out
27 uFL = out(outini: 0, 0, 0)           | u of mold level
28 uFLV = out(outiniV: 0, 0, 1)         | u of motor speed
29 urev = out(urev: 0, 0, 2)            | Rev. of motor speed
END

```

그림 6 Configuration File

피지제어 기능코드로 실험의 결과는 (그림 7)에 나타내었다. (그림 7)의 윗그림은 주조속도를 나타내는데, 실선은 설정치이고 점선은 플랜트의 출력값을 나타낸다. 그림에서보면 설정치를 잘 따라가고 있으므로 피지제어가 잘 구동하고 있음을 알 수 있다. 시간은 Realtime OS인 VxWorks를 사용하여 실시간으로 실험하였다. 아래 그림은 주형레벨을 나타내는 데 그림의 800은 설정치를 나타내고 정상상태로 잘 제어가 되므로 피지제어 기능코드가 실시간으로 제대로 동작하고 있음을 알 수 있다.

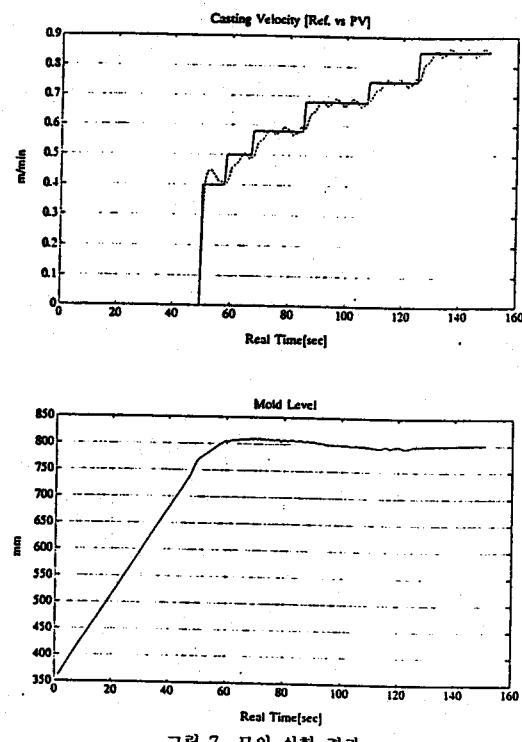


그림 7 모의 실험 결과

5. 결론

본 논문에서는 산업체에서 운용되고 있는 DCS에 한층 진보된 제어알고리즘인 피지제어 알고리즘을 구현하는 방법을 제안하였다. 피지제어를 DCS에 구현하기 위해서는 기능코드의 형태가 되어야 한다. 그 이유는 사용자가 편리하게 제어알고리

즘을 구현하고 제어알고리즘 구현의 일반성을 부여하기 위해서이다.

본 연구에서는 피지제어를 현장의 사용자가 사용하기 편리하도록 기능코드의 형태로 구현하였다.

제어에 사용되는 규칙(Rule)은 자료 파일의 형태로 읽어 들이므로 사용자가 규칙을 변경하거나 추가하려 할 때 쉽게 할 수 있게 하였다. 구현된 기능코드는 범용성을 가지므로 제철소뿐만 아니라 발전소 등의 여러 공정에도 적용이 가능하다. 구현된 기능코드의 검증을 위하여 제철소의 한 공정인 연주공정을 대상으로 하여 피지제어를 수행하였다. 연주공정의 모의실험을 통한 피지제어기의 성능을 평가하기 위하여 대상시스템을 모델링하였다. 연주공정의 주조시작시의 수동운전을 대신할 피지제어기를 구성하였다. 제어에 사용된 규칙은 현장의 조업자들로부터 얻었다. 대상시스템을 실시간 운영체제인 VxWorks로 실시간을 이용한 모의실험을 통하여 구현된 기능코드를 검증하여 기능코드화된 피지제어의 현장에 적용가능성을 보였다.

참고 문헌

- [1] 포함제철 산업기술 연구소(RIST) 이해영,백기남,임동진, "연속 주조 시스템에서의 주형 Level의 Fuzzy 제어", 연구기보 제5권 제2호, 1991 (p277 - 282)
- [2] 住友金屬工業 製鐵技術研究所, "スライティイソクノスルの油壓制御技術の開発", CAMP-ISIJ Vol3(1990) - 1125
- [3] 川崎製鐵 技術研究所本部 千葉製鐵所, "外亂オフサ-ハを用いた連鑄モールト内湯面レヘル 制御", SICE'91 July 17-19 Yonezawa
- [4] 포함제철(주), 철강제품과 생산 공정, (p 401 - 441)
- [5] 포함제철(주) 광양 제철소, 제철 용어집, 1990
- [6] 포함제철(주) 광양 제철소, 연주 공장 설비 사양서, (p 11 - 150)
- [7] 포함제철(주) 광양 제철소, ECLM Sensor 교체 보고서
- [8] 포함제철(주) 광양 제철소 설비관리부 정비기술과, 기술검토서 (연주공장 Tundish Sliding Nozzle 작동검토)
- [9] 이희규,오길록, 피지 이론 및 응용, I, II권
홍릉과학출판사 1991
- [10] Katsuhiko Ogata, Discrete-Time Control System, Prentice-Hall Publishing, 1987
- [11] A.E.Fitzgerald, Electric Machinery, fourth edition
McGraw-Hill Publishing, 1983
- [12] 이희규,문봉체,김병국,변중남 "그래픽 제어 언어를 사용한 공정제어용 다중루프 제어기의 개발" 한국자동제어학술회의논문집 1990.10.26-27 (p 200 - 203)
- [13] 변중남, 김병국, 박동조, 발전소 보일러 제어 시스템용 고급제어 알고리즘 및 고장진단 피지 전문가 시스템의 개발, 한국과학기술원, 1993. 10 (p 1 - 3)