

論文94-31B-7-17

범용의 퍼지 하드웨어 설계

(A Design of the General-Purpose Fuzzy Hardware)

金龍泰*, 李承夏*, 李然定*, 卞增男*

(Yong-Tae Kim, Seung-Ha Lee, Yun-Jung Lee and Zeungnam Bien)

要約

최근들어 수동으로 운전되고 있는 규모가 크고 비선형성이 강한 플랜트를 퍼지 제어를 이용하여 자동화를 많이 하고 있다. 그러나 실시간 처리를 요하는 많은 분야에서는 퍼지 연산 속도의 제한으로 퍼지 제어의 적용에 어려움을 겪고 있다. 실제로 퍼지 제어기는 퍼지화, 퍼지 추론 및 비퍼지화 연산을 수행하기 위해서 많은 시간이 걸리고 있다. 이러한 퍼지 연산 속도 문제를 해결하기 위해서는 새로운 연산 알고리즘을 개발하여 속도를 증가시키든지, 고속의 퍼지 연산 전용의 하드웨어를 이용하는 방법이 있을 것이다. 본 논문에서는 이러한 문제를 해결할 수 있는 범용성 및 확장성을 가지고 있는 새로운 퍼지 하드웨어를 제안하고, 검증을 하겠다.

Abstract

Recently, the fuzzy control is widely used as a tool for constructing automatic control systems which can replace the manual operation of large-scale nonlinear plants. In most applications of the fuzzy control, however, it is hard to meet the requirement of the operation time. In some real-time control, the fuzzy control scheme requires too much computing time for fuzzification, inference, and defuzzification. To reduce the computing time, there may be two alternatives: the development of a new operation algorithm and the design of high-speed fuzzy hardware. In this paper, to solve the problem of reducing the fuzzy operation time, we propose a new high-speed fuzzy hardware scheme which has merits of its generality and extensibility. Finally, we verify the proposed fuzzy hardware.

1. 서론

기존의 주파수 응답을 이용하는 고전적인 제어기 설계 방법이나 상태변수 공간에서의 시스템 기술을

근거로 하는 현대 제어 방법에서는 플랜트의 수학적 모델링이 매우 중요한 역할을 한다. 그러나 매우 복잡한 프로세스는 수학적인 모델을 쉽게 만들 수 없으며, 만들었다 하더라도 적절한 제어기를 설계하기가 쉽지 않다. 또한 이러한 시스템을 모델링했을 경우라도 선형화 작업 등을 통해서 모델링의 오차가 발생하게 된다. 이러한 점으로 말미암아 시스템의 구조 자체가 비선형 특성을 가지고 있거나, 모델링하기 어려

*正會員, 韓國科學技術院 電氣및電子工學科

(Dept. of EE., KAIST)

接受日字: 1993年 7月 29日

운 불명확한 대상, 또는 제어 변수가 많은 선형 시변 시스템과 같은 대상의 제어에는 기존의 제어 방법이 한계를 노출하고 있다. 이러한 수학적 모델을 근거로 한 제어의 한계를 극복하기 위하여, 최근에 컴퓨터의 인공 지능 분야의 발전에 발맞추어 지능 제어에 대한 관심이 고조되고 있으며, 그러한 맥락으로 퍼지 이론과 뉴럴 네트워크에 대한 연구가 많이 진행되고 있다. 이 중에 퍼지 제어는 애매하고 불확실한 정보를 비교적 쉽게 처리할 수가 있는 것으로 선형 시스템은 물론, 비선형 시스템, 모델링 작업이 어려운 복잡한 시스템의 제어에 매우 유용하다는 것이 알려져 있다. 이러한 퍼지 제어의 특징은, 첫째 인간의 정확한 지식뿐만 아니라 애매한 지식을 이용할 수 있기 때문에 인간과 비슷한 판단을 내릴 수 있다는 것이고, 둘째는 비교적 인간의 간단한 지식을 사용하기 때문에 쉽고 저렴하게 적용이 가능하며, 셋째로 복잡하고 이해하기 힘든 시스템의 제어에의 적용도 가능하다는 것으로 요약될 수 있다.

현재 퍼지 이론은 제어 분야에서 가장 활발히 응용되고 있으며, 그 외에도 공학계(엔지니어링 설계, 신호처리, 패턴인식, 이상진단), 사회계(의사결정, 행동과학, 경제 사회모델), 자연계(기상, 생태계, 물리 화학적 현상규명) 등 다방면에 걸쳐 유용하게 응용될 수 있다는 것이 입증되어 활발히 연구 중에 있다.¹⁾ 그러나 퍼지 제어가 많은 분야에 적용이 되고 있음에도 불구하고, 퍼지 연산이 기존의 연산과는 다르게 표현되기 때문에 컴퓨터로, 즉 소프트웨어적인 방법으로 퍼지 추론을 실시간 모드에 수행하기가 쉽지 않다. 이러한 이유로 대규모의 공정 제어, 산업 기계, OAG기, 통신, 고급 가전 제품 등 퍼지 연산이 대규모로 실행되면서도 실시간성이 요구되는 분야에는 아직까지 적용에 제한이 있다. 그러므로 퍼지 연산 전용의 추론 하드웨어의 개발이 요구되고 있으며, 이에 부응하여 많은 연구가 진행되고 있다.

앞에서 언급한 바와 같이 퍼지 이론을 이용한 수많은 응용 사례가 있음에도 불구하고, 아직까지 대규모의 퍼지 연산의 수행 및 실시간 제어가 요구되는 시스템의 경우는 기존의 소프트웨어적인 퍼지 제어 방식을 적용할 수가 없다. 특히, 현대의 대규모의 공정 제어 시스템이 대부분 많은 입력과 출력을 갖는 다변수 거대 시스템이기 때문에 기존의 소프트웨어적인 방법으로 퍼지 제어를 적용하고자 할 경우는 수많은 제어 규칙을 저장해야 하므로 메모리의 부족이라는 문제가 발생하고, 프로그램의 크기도 방대하고 번잡하므로 소프트웨어에 의한 직렬 처리로는 퍼지 제어의 적용이 불가능하게 된다. 또한 기존의 입력과 출

력의 확장성을 갖지 않는 퍼지 하드웨어도 마찬가지로 적용이 불가능하게 된다. 본 논문에서는 이러한 문제의 해결 방안으로 확장성, 범용성 및 고속성을 갖는 새로운 퍼지 하드웨어를 제안하겠다. 이러한 퍼지 하드웨어를 이용한다면, 첫째로 대규모의 퍼지 추론을 요구하고 실시간성이 요구되는 분야는 물론이고, 많은 입력과 출력을 갖고 있는 시스템에 고속 추론의 도구가 제공됨으로써 퍼지 제어를 적용할 수 있을 것이고, 둘째로는 대부분의 응용 분야에 퍼지 제어 규칙만을 수집해 적용이 가능하므로 개발 기간의 단축, 쉬운 적용 등의 효과를 얻을 수 있다.

본 논문의 구성은 다음과 같이 구성되어 있다. 먼저 II장에서는 퍼지 하드웨어의 개요, 구현 방법, 그리고 특징을 기술하였고, III장에서는 제안된 퍼지 하드웨어를 검증하고 IV장에서 결론을 맺었다.

II. 퍼지 하드웨어의 설계

1. 퍼지 하드웨어의 개요

퍼지 제어기는 일반적으로 퍼지화기(fuzzifier), 퍼지 추론기(fuzzy reasoning engine), 비퍼지화기(defuzzifier)의 3부분으로 구성된다. 퍼지화기는 입력 변수들의 값을 측정하여 애매하지 않은(crisp) 입력값을 퍼지 추론을 행할 수 있는 퍼지 집합으로 변환시키는 역할을 한다. 퍼지화 방법에는 소속함수의 모양에 따라서 싱글톤 방법과 이등변 삼각형 방법 등이 있다. 퍼지 추론 방법에는 직접법, 간접법, 하이브리드 방법과 같이 크게 3가지의 부류가 있으며, 직접법은 다시 Mamdani의 방법(max-min), Larsen의 방법(max-product), Mizumoto의 방법(bounded sum-product)이 있다. 비퍼지화기는 퍼지 추론에 의한 애매한 결과를 애매하지 않은 하나의 수치로 변환하는 역할을 하며 추론 결과의 해석이라고 말할 수 있다. 비퍼지화 방법에는 최대값 방법(maximum criterion), 최대값 평균 방법(mean of maximum, MOM), 그리고 면적 중심법(center of gravity, COG) 등이 있다.

퍼지 추론을 통해 제어기 출력값을 구하는 방법은 크게 범용 컴퓨터에서 소프트웨어적으로 처리하는 방법과 하드웨어를 이용하는 방법이 있다. 소프트웨어로 퍼지 추론 출력을 구할 때는 퍼지 추론의 정밀도(resolution)가 세밀할수록, 제어기의 입, 출력 갯수가 많아질수록, 추론에 걸리는 시간은 그에 비례해서 현저하게 길어지게 된다. 따라서 퍼지 추론 과정을 빠르게 처리할 수 있는 전용 하드웨어의 개발이 세계적으로 많이 진행되고 있는데, 이는 퍼지 추론 과정

이 하드웨어화하기 쉽고 병렬 처리를 통한 추론 시간의 감소를 피할 수 있는 장점을 갖고 있기 때문이다. 최근까지 퍼지 논리 추론의 전용 하드웨어 개발 연구는 Yamagawa, Watanabe, Togai의 연구를 시작으로 최근까지 OMRON(주), OKI(주), HITACHI(주), 삼성 항공(주), KAIST 등 여러 곳에서 진행되었다.

1987년 규슈 공대의 Yamakawa 교수가 개발한 퍼지 칩은 전형적인 아날로그 방식의 퍼지 추론 칩으로 널리 알려져 있다.^[2] Yamakawa 칩은 규칙 칩(rule chip)과 비퍼지화 칩(defuzzifier chip)으로 구성되어 있으며, 규칙 칩은 그 하나가 입력 변수 3개, 출력 변수 1개인 제어 시스템의 제어 규칙 1개를 처리할 수 있다. 이러한 Yamakawa 칩은 아날로그 방식이므로 속도가 비교적 빠르고 확장 방식이 결선(wired-max) 방식으로 간단한 반면, 노이즈가 전압에 미치는 영향이 추론의 정확성에 영향을 미치며, 하나의 규칙 칩이 하나의 규칙만을 처리하므로 많은 규칙이 필요한 제어 시스템에는 여러개의 규칙 칩이 연결되어야 하며, 디지털 제어 시스템과의 인터페이스가 어려운 단점이 있다.

일반적으로 현재 개발중인 추론 전용의 퍼지 하드웨어는 이와같은 아날로그 방식보다는 디지털 방식의 퍼지 추론기가 주류를 이루고 있다. 1986년 AT&T 벨 연구소의 Togai, Watanabe가 개발한 근사 추론 엔진은 디지털 방식의 퍼지 추론 시스템의 전형적인 모델로 여겨지고 있다.^[3] 이 추론 엔진은 퍼지 소속함수 값을 32개의 4비트(bit) 디지털 값으로 표현하며, 규칙 회로에서 보면 4개의 입력이 규칙과 최소-최대(min-max)회로를 거치면 적합도를 구하게 되고, 4개의 적합도의 최소값을 취하면 최종 적합도를 얻게 되고, 규칙의 후진부값들과 이 최종 적합도에 의해 컷트된 출력 소속함수를 얻을 수 있다. 이렇게 구한 출력 소속함수는 비퍼지화 회로에 의해 출력이 계산된다. 이 디지털 VLSI 칩은 초당 약 58만번의 추론을 할 수 있으며, 4 입력 2 출력의 51개 규칙, 혹은 2 입력 1 출력의 102개 규칙의 두가지 모드로 동작이 가능하게 되어 있다. 그러나 이 칩이 입출력 변수를 늘리기 위해서는 트리(tree) 형태로 연결을 해야 하기 때문에 확장성이 용이하지 못하고 규칙 확장은 할 수 없는 단점이 있다. 또한 소속함수의 정밀도도 32×4비트로 떨어지는 편이다. 이러한 칩들이 시작품이라고 한다면, FC110 디지털 퍼지 프로세서는 Togai Infra Logic사에서 개발한 프로세서 형태의 칩으로 칩, 혹은 칩을 장착한 가속 보드 형태로 판매되고 있는 시스템이다. 이 칩의 특징은 8비트로

데이터를 처리하며 초당 10만 추론이 가능하며 규칙은 800개, 입출력 수는 각각 256개까지 가능하다는 것이지만, 허용 입 출력 변수의 개수가 한정되어 있고 확장이 불가능하게 되어 있다는 것이 단점으로 지적되고 있다. 그외에도 많은 사람에 의해서 퍼지 하드웨어에 대한 연구가 활발히 진행되고 있다.

우리 나라에서는 한국과학기술원에서 만든 Kafa(KAIST Fuzzy Accelerator)^[4]와 삼성항공(주)에서 PLC(Programable Logic Controller)의 한 모듈로 퍼지 모듈을 개발한 것^[5]을 제외하면 선례가 거의 없다. 앞의 선례 연구에서와 같이 많은 퍼지 추론 전용 시스템이 개발되고 있으나, 입출력을 용이하게 확장할 수 있는 확장성과 여러가지의 추론 방법과 비퍼지화 방법을 수행할 수 있는 범용성을 갖는 시스템은 찾기 어렵다. 즉 하나의 단위 시스템이 몇 개의 입출력에 대해서 처리하게 되어 있고, 그러한 단위 시스템을 여러개 연결하면 많은 입출력의 처리가능한 시스템이 있다면, 이는 작은 응용 분야는 물론이고 많은 입출력이 관계되는 대규모 시스템까지 적용될 수 있는 장점을 지니게 된다. 이러한 사실을 바탕으로 제안된 추론 하드웨어는 범용성, 확장성을 가지면서도 퍼지 추론의 장점을 십분 활용하여 최적화된 병렬 처리 방식을 적용하고 있다. 퍼지 제어 시스템의 구조는 그림 1과 같이 마스터 CPU 모듈과 퍼지 하드웨어 부분으로 나뉘는데, 여기서 퍼지 하드웨어 부분은 퍼지화된 입력을 받아서 미리 설정된 규칙과 소속 함수에 의해 비퍼지화된 추론출력을 구하는 과정을 맡고 있다. 즉, 퍼지 하드웨어는 마스터 CPU에서 보면 퍼지 코프로세서처럼 동작하게 된다. 그림 2는 IF, THEN 모듈로 개발된 퍼지 추론 하드웨어의 모습이다.

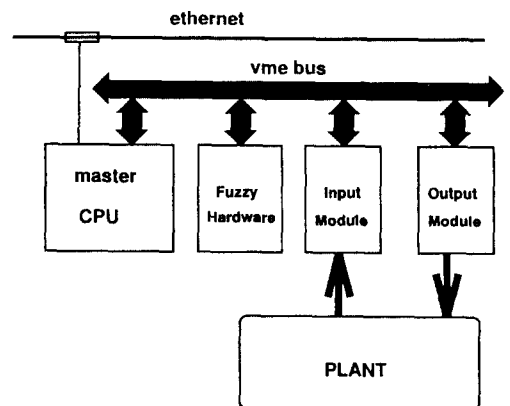


그림 1. 퍼지 제어 시스템의 구조
Fig. 1. Structure of fuzzy control system.

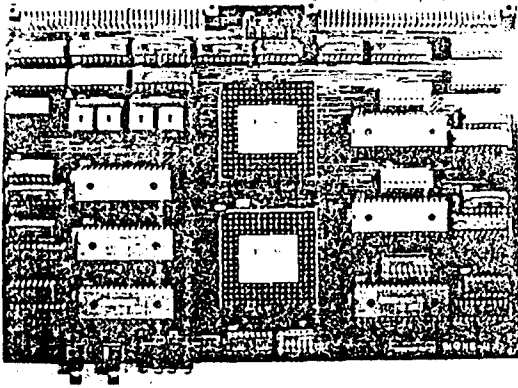


그림 2. 퍼지 하드웨어의 사진
Fig. 2. Photograph of fuzzy hardware.

2. 퍼지 하드웨어의 구현 방법

1) 퍼지 하드웨어 시스템의 자료 구조

퍼지집합은 소속함수로 정의되는데, 일반적으로 이 함수의 모양은 매우 다양한 형태의 연속함수이다. 그러나 다양한 형태의 연속함수를 퍼지 하드웨어에서 다루기는 상당한 어려움이 따르므로, 많은 경우에 전체 집합 내에서 m개의 원소를 샘플링하여 이 원소들의 소속정도를 표시함으로서 소속함수를 나타낸다. 그림 3과 같이 퍼지집합은 m개의 원소를 가지는 벡터로 표현할 수 있다.⁶⁾

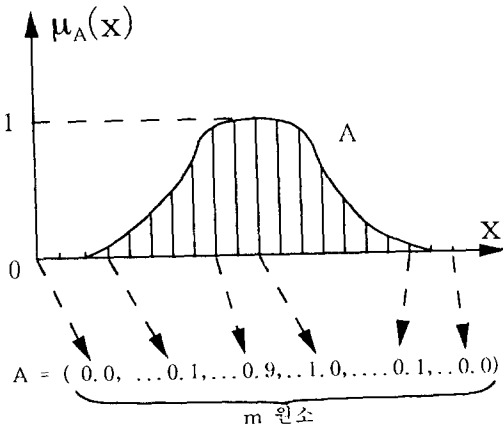


그림 3. m원소 퍼지 집합에 의한 퍼지 집합의 표현
Fig. 3. Discription of fuzzy set using m-element fuzzy set.

이와 같이 m개의 원소로 나타낸 퍼지집합을 m원소 퍼지집합이라 부르고 있다. 실제 이 값들은 퍼지 하드웨어에서 양자화(quantization)된 값으로 표시

할 수 있다. 제안된 퍼지 하드웨어는 256원소 퍼지 집합을 사용하며, 0과 1사이의 소속함수 값은 최대 256개까지 양자화가 가능하도록 설계가 되었다.

2) 퍼지 하드웨어의 개념설계

앞에서 언급한 바와 같이 제안하는 퍼지 하드웨어는 범용성, 확장성 및 고속성을 고려하였다. 먼저 범용성을 고려하기 위해서 먼저 일반적으로 퍼지 제어에 많이 사용되고 있는 직접추론방법들, 즉 max-min, max-product, bounded sum-product의 3가지 방법이 수행되도록 하였고, 비퍼지화방법에 있어서도 일반적으로 많이 사용되는 방법들, max criterion, MOM, COG을 수행할 수 있도록 구성하였다. 또한 임의의 모양의 소속함수를 사용할 수 있게 설계되었다.

퍼지 추론 과정을 통해 제어기 출력을 구하는 과정은 max-min 추론방법의 경우 그림 4와 같이 나타나어진다. 이러한 과정을 그림 4의 아래 부분에 보인 것과 같이 각 기능별로 IF, THEN 모듈로 나누고 각 모듈간 연결 및 동작을 정의하여 퍼지 하드웨어를 구성하였다. 퍼지화된 입력이 IF 모듈로 전해지면 일련의 동작을 통해 적합도들의 최소값을 구하고, 다음 모듈은 그 값을 이용해서 추론을 수행하면 그 결과가 THEN 모듈의 메모리에 출력 소속 함수가 갱신되어 저장된다. 이러한 전달 및 계산 과정을 제어 규칙 갯수만큼 반복하면 최종의 출력 소속 함수가 THEN 모듈에 저장되고, 이를 비퍼지화하면 제어기 출력을 얻는다.

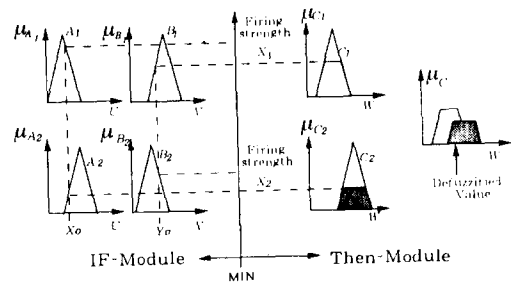
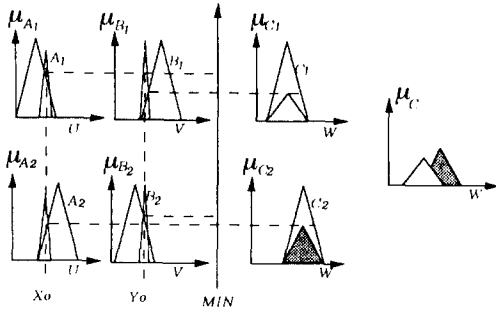


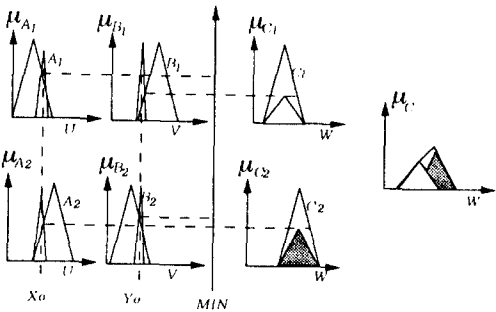
그림 4. 퍼지 추론의 과정
Fig. 4. Procedure of fuzzy inference.

그외의 일반적인 직접 추론 방법은 그림 5와 같이 표현된다. 그림 4와 그림 5에서 볼 수 있는 것과 같이 세가지 방법에 있어 몇가지 특징이 있다는 것을 알 수 있다. 먼저 가운데 최소값 연산(min)직선의 왼편은 세가지 방법 모두 같다는 것이며, 오른편은

추론 방법별로 약간의 차이는 있지만, 유사한 형태를 가지고 있다는 것을 알 수 있다. 또한 가운데 최소값 연산 직선의 왼편은 퍼지규칙에서 IF부분(전건부)에만 관련이 되어 있으며, 오른편은 Then부분(후건부)에만 관련되어 있다는 것을 알 수 있다.



(a)



(b)

그림 5. 직접 추론 방법(max-product, bounded sum-product)

Fig. 5. Direct inference method. (max-product, bounded sum-product)

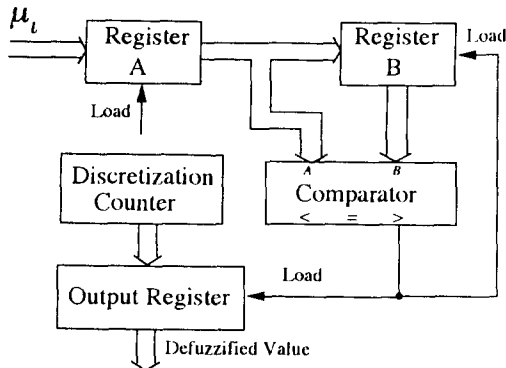
퍼지 하드웨어에는 소속함수 및 규칙이 저장되어야 하는 데, 규칙의 전건부와 후건부를 따로 구분함으로써 많은 규칙을 저장할 수 있으며, 입출력변수를 확장할 때는 각 모듈에서 만들어진 소속정도의 최소값을 다른 모듈에 전달함으로써 쉽게 확장이 가능하다. 앞으로 이 소속 정도의 최소값을 발화정도값(firing strength)라고 명명 하겠다. 또한 IF부분과 Then부분을 구분함으로써 규칙별로 파이프라인(pipe-line) 처리가 가능하다. 각 입력변수에 대한 적합도의 계산은 제어기에서 퍼지화 방법으로 싱글톤 방법을 사용

하기 때문에 메모리의 맵핑을 이용해 구현하였으며, 이렇게 함으로써 속도를 빨리할 수가 있다. 각 모듈에서 발화정도 값을 계산하는 것은 다른 모듈에서 오는 발화정도 값도 같이 최소값을 계산하여 입력변수의 확장을 가능하게 하였다. 또한 IF 모듈에서 계산된 발화정도 값을 조사해서 0이 되는 경우는 많은 연산이 있는 THEN 모듈에서 계산을 하지 않아도 되므로 이 발화정도값을 이용해 추론 속도를 증가시킬 수 있다.

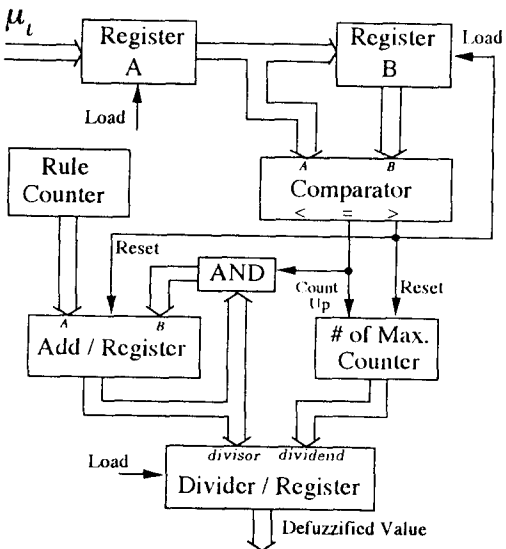
직접 추론 방법 3가지를 모두 구현하기 위해서는 min회로와 product 회로, max 회로와 bounded sum 회로 중에 각각 한가지씩을 선택해서 임의적으로 세가지방법 중 한가지방법을 선택하도록 하였다. 여기서 min, max, bounded sum 회로는 구성이 비교적 간단하나, product 회로는 구현하기가 까다로운 회로이다. product 회로를 구현하는 방법은 일반적으로 조합(combination)회로와 순차(sequential) 회로를 이용하여 구성된 회로가 이용된다. 본 연구에서는 시간상으로 빠르고, 또한 전체 제어로직 외에 곱셈기의 제어로직이 사용되지 않는 방법으로 단지 조합소자들을 사용하는 Braun에 의해 제안된 반복형 단위 배열 곱셈기(iterative cellular array multiplier)를 사용하였다.⁷⁾ 이 방법은 비교적 많은 조합소자가 사용되지만 계산시간이 빠르고 순차적인 제어로직이 사용되지 않는 점에서 구현하기 쉽다.

일반적으로 많이 사용되는 비퍼지화 방법으로 maximum criterion, MOM, COG 등이 있다. 이 세가지 비퍼지화 방법의 구현 알고리즘은 그림 6과 같다.

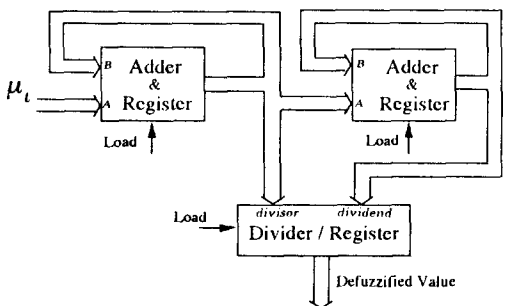
Maximum criterion방법은 추론된 결과의 전체 집합(universe of discourse)을 마지막 이산화 단계(discretization level)에서 부터 처음 단계까지 최대값을 찾아내는 것이고, MOM은 이렇게 찾아진 최대값들의 평균값을 찾아내는 것이다. COG방법은 덧셈기와 나눗셈기로 쉽게 구현하는 방법을 사용하였다.³⁾ 여기서 나눗셈기를 구현하는 방법으로는 Guild가 제안한 비저장형 배열 나눗셈기(non-restoring array divider)를 사용하였다.⁷⁾ 계산속도가 빠르고 별도의 제어순차회로가 사용되지 않고 구현하기 쉬운 장점이 있다. 여기서 발화정도와 이산화 단계가 각각 8비트로 표현되기 때문에 나눗셈기는 피셋수가 23비트, 켓수가 16비트로 구성된다. 퍼지 하드웨어는 메모리를 제외한 대부분 회로가 FPGA(field programmable logic array) 기술로 된 프로그래밍 가능한 로직 회로인 XILINX사의 XC3090-70에 구현하였다.



(a)



(b)



(c)

그림 6. 비퍼지화 방법(max. criterion, MOM, COG)

Fig. 6. defuzzification method.(max. criterion, MOM, COG)

3) IF 모듈의 설계

IF모듈은 앞서서도 언급한 바와 같이 각 규칙에서 발화정도를 계산해 내는 기능을 가지며, 크게 3가지 부분-VME 버스 인터페이스부, 제어로직부, 퍼지 디코더(fuzzy decoder)부-으로 구성된다. 그림 7에 IF 모듈의 구성을 보였다.

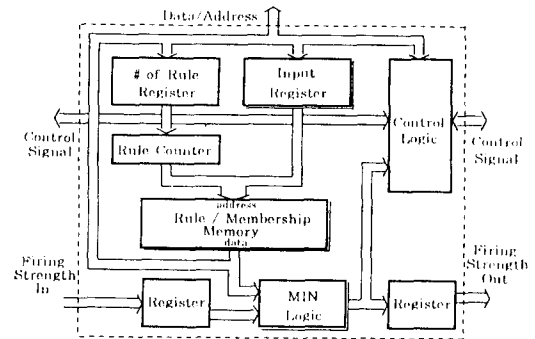


그림 7. IF 모듈의 구조

Fig. 7. Structure of IF module.

여기서 VME 버스 인터페이스부는 일반적으로 공정 제어 시스템에서 많이 사용되는 VME 버스를 통해 마스터 CPU로부터 제어기의 입력을 받아들이고, 또한 규칙과 소속함수메모리에 원하는 형태의 데이터를 마스터 모듈로부터 전송받기 위해서 사용된다. 제어로직부분은 별도의 CPU를 사용하지 않고 플립플롭(flip-flop)을 이용하여 구성하였다. 먼저 제어 흐름도(control flow diagram)를 구성하고 이것을 SM(State Machine) 차트로 변환하고 이 차트로부터 직접 플립플롭과 논리소자를 이용하여 제어로직을 구성하였다.^[8] 퍼지 디코더는 각 변수값의 입력으로부터 적합도를 계산하고 이 값의 최소값을 계산하는 것으로, 앞의 추론 방식에서 언급했던 것처럼 메모리의 맵핑을 이용하여 적합도를 계산하고, 이 값들의 최소값을 계산하여 나온 발화정도 값을 다음의 IF 모듈, 또는 Then 모듈에 전달한다.

4) Then 모듈의 설계

Then 모듈은 IF 모듈에서 계산된 발화정도 값을 받아 실제로 추론하고 비퍼지화하는 기능을 가지고 있으며, 3가지부분-VME 버스 인터페이스부, 제어로직부, 추론 및 비퍼지화부-로 구성된다. Then 모듈의 구성은 그림 8과 같다.

VME 버스 인터페이스부는 비퍼지화가 끝나면 마스터 CPU에 인터럽트를 사용해 계산 결과를 전달해

주고 또한 마스터 CPU로부터 규칙과 소속함수의 원하는 형태를 메모리에 저장하기 위해 전송받기 위해서 사용된다. 제어로직부는 IF 모듈과 같은 방법으로 설계되었다. [8] THEN 모듈의 추론및 비퍼지화부는 앞의 전체 개념에서 설명한 것과 같이 3가지의 직접 추론법과 3가지의 비퍼지화법 중 각각 한 방법씩을 선택하여 수행할 수 있도록 구성하였다. 이 부분에서는 이산화(discretization) 수만큼의 추론된 값을 저장한 임시적인 메모리가 있어야 하는데, 이부분에는 FIFO(first-in first-out) 메모리를 사용해 속도도 증가시키고, 회로도 간단하게 구현하였다.

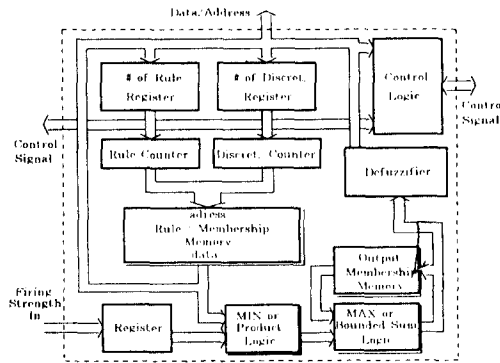


그림 8. THEN 모듈의 구조
Fig. 8. Structure of THEN module.

3. 퍼지 하드웨어의 특징

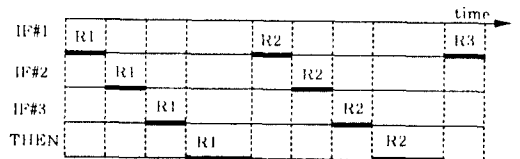
앞에서 설명한 추론 하드웨어의 동작을 시간적으로 나타내면 그림 9(a)와 같다. 각 모듈(IF, THEN)이 규칙에 대한 처리를 제어 규칙 수 만큼 처리하고 난 후에 비퍼지화를 행하게 되어 있다. 이러한 방식의 동작을 IF모듈 1과 2를 각기 다른 규칙의 처리에 대해서 병렬로 처리하면 그림 9(b)와 같고 이렇게 하면 처리 속도를 많이 줄일 수 있다. 퍼지 추론의 특징인 규칙 처리의 병렬 처리 가능성을 이용한 것이다. 대부분의 경우에 퍼지 제어기에서 제어 규칙이 제어 입력에 대해 모두 그 적합도가 0보다 큰 경우는 발생하기가 어렵다. 다시 말하면, 어떤 제어 입력에 대해서 대부분의 제어 규칙들은 그 발화정도가 0이 되고, 단지 두세개의 규칙만이 0이 아니게 되므로 발화정도가 0인 규칙들에 대해서는 THEN 모듈의 처리가 필요 없게 된다. 이러한 사실을 이용하면 추론 처리 시간을 줄일 수 있는데 이렇게 되면 각 모듈별 동작은 그림 9(c)와 같이 나타난다. 즉, 규칙 2번은 적합도가 0이므로 그 처리 시간이 매우 짧게 줄었다. 본 연구에서 개발한 추론 하드웨어에는 이와같은 병렬 처리

및 최적화 처리 방식이 적용되어 우수한 확장성을 가지면서도 확장에 따른 추론 속도의 저하를 최대한으로 줄일 수 있도록 설계되었다. 현재 제작된 퍼지 하드웨어의 추론 속도는 다음 식과 같이 클럭의 개수로 계산된다.

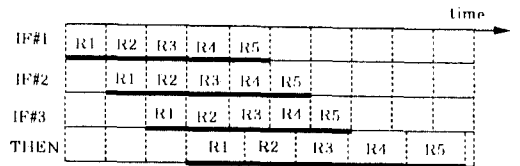
$$13 \times M + 3 \times N + 6$$

(N: 규칙의 수, M: 출력 멤버십의 이산화 수)

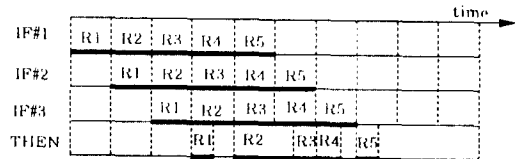
예를 들어, 10MHz의 클럭을 사용했을때 일반적인 2입력, 1출력, 25개의 규칙에 대해서 32이산화 단계를 가지는 퍼지 제어기를 구성했을 경우, 제작된 퍼지 하드웨어는 약 20KFLIPS(fuzzy logic inference per second)의 속도를 가지고 있다.



(a)



(b)



(c)

그림 9. 최적화된 병렬처리를 이용한 퍼지 추론
Fig. 9. Fuzzy inference using optimized parallel processing.

이상의 내용을 요약하면, 제작된 퍼지 하드웨어는 기존의 퍼지 하드웨어와는 다르게 다음과 같은 방법으로 범용성, 확장성, 고속성을 실현하고자 하였다. 첫째로, 범용성을 위해 3가지 추론법과 3가지 비퍼지화방법 중에 각각 한 방법씩을 선택하여 사용할 수 있도록 구성하였으며, 임의로 소속함수의 모양과 256 단계 이내에서 입출력 해상도를 정하도록 하였다. 둘째로, 확장성을 위해서 I/O변수 및 규칙의 확장이 가

능하도록 설계하였다. 셋째로, 고속성을 위해서 적합도 계산을 메모리 맵핑 방법으로 하였으며, IF, THEN 모듈에서 규칙별로 파이프라인 처리를 하고 있으며, 또한 앞에서 언급한 바와 같이 최적화된 병렬처리를 사용해 속도를 증가시키고 있다. 본 연구에 의해 제작된 퍼지 하드웨어의 성능은 표 1과 같다.

표 1. 제안된 퍼지 하드웨어의 성능
Table 1. Performance of the proposed fuzzy hardware.

항 목	특 성	항 목	특 성
퍼지화 방법	Singleton 방법	입출력 정밀도	256(조정가능)
입력 변수 수	8(확장가능)	출력 변수 수	4(확장가능)
최대 추론 속도	약 20KFLIPS	최대 규칙 수	256(확장가능)
추론법	Max-Min법 Max-Product법 Sum-Product법	비퍼지화법	Max. Criterion MOM COG

Ⅲ. 퍼지 하드웨어의 검증

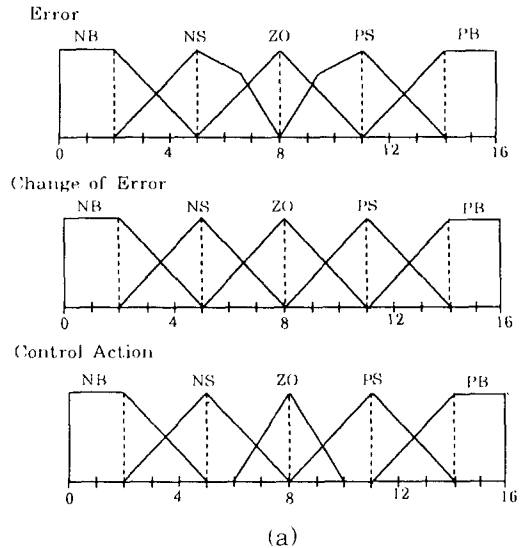
1. 전체 시스템의 구성

퍼지 하드웨어를 이용한 퍼지 제어 시스템의 구성은 앞에서 제시한 그림 1과 같이 마스터 CPU 모듈 및 퍼지 하드웨어로 구성되며, 입출력 모듈을 통해 플랜트와 연결된다. 여기서 퍼지 하드웨어 부분은 퍼지화된 입력을 받아서 미리 설정된 규칙과 소속 함수에 의해 추론 출력을 구하는 과정을 맡고 있다. 즉, 퍼지 추론 하드웨어 부분은 마스터 CPU의 퍼지 코 프로세서 형태로 동작하게 된다. 마스터 CPU 모듈은 제어가 자율적인 동작을 할 수 있도록 입력과 출력 모듈에 제어량을 쓰거나, 현재플랜트의 출력을 읽어오는 일, 퍼지 하드웨어 모듈에 퍼지화된 입력을 주고, 퍼지 추론된 출력을 얻어내는 일을 맡고 있다. 마스터 CPU 모듈로는 모토롤라 MC68030을 CPU로 탑재한 FORCE사의 CPU30 보드를 사용하고 있으며, 오퍼레이팅 시스템(operating system)으로 실시간 오퍼레이팅 시스템인 VxWorks를 이용하였다. 마스터 CPU 모듈, 퍼지 하드웨어, 그리고 입출력 모듈은 VME 버스를 통해서 통신이 이루어지게 되어 있다.

2. 퍼지 하드웨어의 검증

앞에서 언급한 바와 같이, 퍼지 하드웨어는 세가지 추론법과 세가지의 비퍼지화 방법을 수행하도록 설계

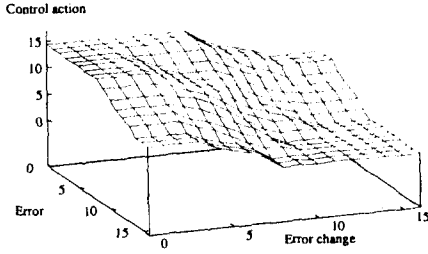
되었는데, 각 추론법과 비퍼지화법에 대해서 퍼지 하드웨어를 이용해 계산한 결과를 그림 11에서 그림 15까지 나타냈다. 이것은 그림 10에서 보이는 것과 같은 2입력, 1출력의 제어기에 대해서 17개의 이산화 수를 가지고 검증하였다. 검증을 위해 먼저, 컴퓨터로 시뮬레이션을 하여 그 결과(그림 11(a))를 먼저 보고, 똑같은 조건하에 퍼지 하드웨어로 추론한 결과(그림 11(b))를 룩업 테이블(look-up table)방식을 이용하여 비교하였다.^[8] 결과에서 알 수 있듯이 양자화에 따른 약간의 오차는 있으나 알고리즘이 하드웨어로 잘 구현되었다는 것을 알 수 있다. 또한, 본 논문에서 제안된 퍼지 하드웨어는 많은 입출력을 가지고 있는 보일러 시스템의 기동 및 정지의 자동화를 위한 퍼지 제어 시스템에 사용되어 그 유용성을 보여주었다.^[9]



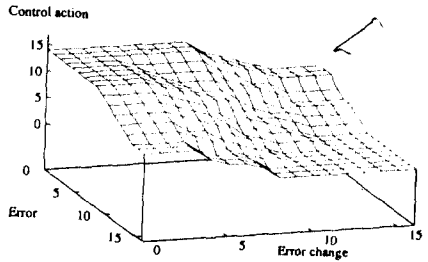
Change of Error / Error	NB	NS	ZO	PS	PB
PB	NB	NB	NB	NS	ZO
PS	NB	NB	NS	ZO	PS
ZO	NB	NS	ZO	PS	PB
NS	NS	ZO	PS	PS	PB
NB	ZO	PS	PB	PB	PB

(b)

그림 10. 검증용 퍼지제어기의 소속함수 및 규칙
Fig. 10. Membership function and rules of fuzzy controller.



(a)



(b)

그림 11. 퍼지 하드웨어의 검증(max-min, COG)
Fig. 11. Verifaion of fuzzy hardware(max-min, COG).

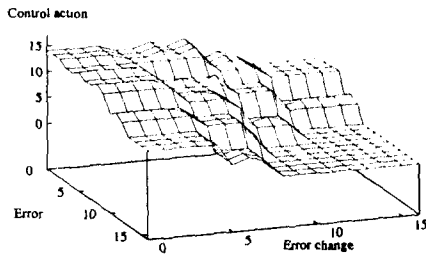


그림 12. 퍼지 하드웨어의 검증(max-min, max. criterion)

Fig. 12. Verifaion of fuzzy hardware(max-min, max. criterion).

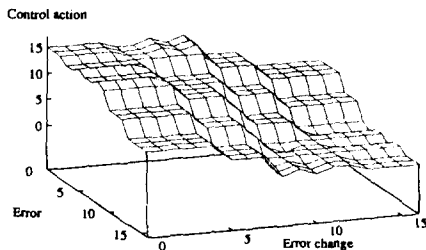


그림 13. 퍼지 하드웨어의 검증(max-min, MOM)
Fig. 13. Verifaion of fuzzy hardware(max-min, MOM).

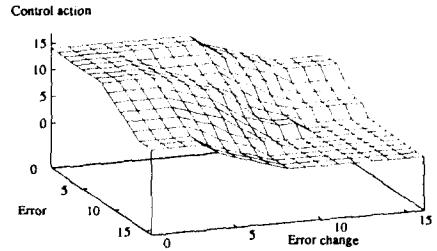


그림 14. 퍼지 하드웨어의 검증(max-product, COG)

Fig. 14. Verifaion of fuzzy hardware(max-product, COG).

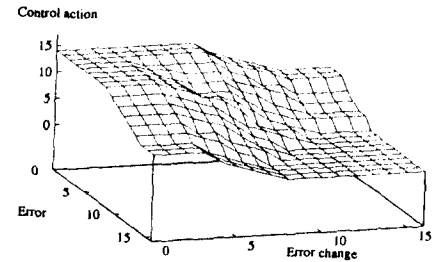


그림 15. 퍼지 하드웨어의 검증(bounded sum-product, COG)

Fig. 15. Verifaion of fuzzy hardware (bounded sum-product, COG).

IV. 결론 및 추후 과제

본 논문에서는 먼저 대규모의 퍼지 연산이 수행되면 서도 실시간성이 만족되는 퍼지 제어를 구현하기 위 해 절실히 요구되고 있는 새로운 범용의 퍼지 하드웨어 를 설계 및 제작하고 검증하였다. 제작된 퍼지 하드웨어 는 먼저 다양한 직접추론법과 비퍼지화 방법을 제공 함으로서 범용성을 만족하도록 제작되었으며, 많은 입 출력을 갖는 시스템에도 적용할 수 있도록 입출력의 확 장을 고려하여 확장성을 가지도록 설계하였다. 본 논문 의 결과로 제작된 퍼지 하드웨어를 이용한다면, 첫째, 대규모의 퍼지 추론을 요구하고 실시간성이 요구되는 분야뿐 아니라, 많은 입출력을 가지고 있는 시스템에도 고속 추론의 퍼지 제어를 적용할 수 있을 것이고, 둘째 로 대부분의 응용분야에 퍼지 제어 규칙만을 수집해 적용이 가능하므로 개발 기간의 단축 및 손쉬운 적용 등 의 효과를 얻을 수 있을 것이다. 그러나 본 논문에서 제안된 퍼지 하드웨어는 이미 상품화된 제품과 비교할 때, 아직 추론속도 면에서 개선을 요구하고 있다. 이러 한 문제는 현재 보드레벨로 제작된 하드웨어를 원칩화

하던지, ASIC(application specific integrated circuit)화하면 개선될 것이다.

參 考 文 獻

[1] 本多中二, 大里有生, *퍼지공학 입문*, 해문당, 1989.

[2] T. Yamagawa, "Fuzzy Microprocessors - How It Works." *Proceedings of KFMS Spring Conference*, vol. 1, no. 1, 1991.

[3] H. Watanabe, "A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture." *IEEE Journal of Solid-state Circuits*, vol. 25, no. 2, April, 1990.

[4] Y. D. Kim, K. H. Lee and K. H. Park, "Parallel Fuzzy Information Processing System KAFA : KAist Fuzzy Accelerator." *fifth IFSA world congress*, Vol 2, 1993.

[5] 최성국, 김영준, 박희재 등, "고속 퍼지 추론 엔진 개발과 산업용 프로그래머블 콘 트롤러에 의 응용." *Proceedings of KFMS Spring Conference*, vol. 1, no. 1, 1991.

[6] 이광형, 오길록, *퍼지 이론 및 응용 II 권 : 응용*, 홍릉과학출판사, 1992.

[7] Kai Hwang, *Computer Arithmetic Principles, Architecture, and Design*, John Wiley & Sons Inc., 1979.

[8] 김용태, *범용 퍼지 하드웨어의 설계 및 보일러 시스템의 자동 기동 정지에의 응용*, 한국과학기술원, 1993.

[9] Zeungnam Bien et. al, "Development of a Fuzzy Control System for Industrial Processes." *Proceedings of first Asian Control Conference*, Tokyo, 1994. (to be published)

著 者 紹 介



金 龍 泰(正會員)

1969年 1月 15日生. 1991年 2月 연세대학교 전자공학과 졸업(공학사). 1993年 2月 한국과학기술원 전기및 전자공학과 졸업(공학석사). 1993年 3月 ~ 현재 한국과학기술원 전기및전자공학과 박사과정 재학중.

주관심 분야는 퍼지제어, 지능제어, 산업자동화 등임.



李 承 夏(正會員)

1965年 8月 17日生. 1988年 2月 경북대학교 공과대학 전자공학과 졸업(공학사). 1990年 2月 한국과학기술원 전기및전자공학과 졸업(공학석사). 1990年 3月 ~ 현재 한국과학기술원 전기및전자공학과

박사과정 재학중. 주관심 분야는 로보틱스, 퍼지시스템, 지능제어 등임.



李 然 定(正會員)

1961年 9月 10日生. 1984年 2月 한양대학교 전자공학과 졸업(공학사). 1986年 2月 한국과학기술원 전기및전자공학과 졸업(공학석사). 1986年 3月 ~ 1989年 8月 한국기계연구원 연구원. 1994年 2月 한국과학기술원 전기및전자공학과 졸업(공학박사).

1994年 3月 ~ 현재 한국과학기술원 정보전자연구소 연수연구원. 주관심 분야는 퍼지제어, 보행로봇제어, 산업자동화 등임.

卞 增 男(正會員) 第 30卷 B編 第 10號 參照

한국과학기술원 전기및전자공학과 교수