

언어변수 및 계수선택방법을 이용한 퍼지제어기 설계에 관한 연구

(A Study on the Construction method to improve
the fuzzy controllers using language variable and
coefficient selecting method)

박승용, 변기녕, 황종학, 김흥수
인하대학교 전자공학과

Seung-Yong Park, Gi-Noung Byun, Jong-Hak Hwang, Heung-Soo Kim
Dept. of Electronic Eng. In-Ha University

요 약

본 논문에서는 제산기가 제거된 싱글톤 퍼지제어기에 대한 새로운 귀속도 함수 발생기 및 계수 선택기 회로를 제안함으로써 회로 구성소자를 줄일 수 있는 기법을 제안하였다. 입력값의 범위에 따라 추론에 사용될 언어변수를 미리 결정하여 필요한 귀속도 함수만을 발생시킴으로서 보다 간략화된 회로를 구성할 수 있으며 또한 추론의 결과에 영향을 줄 계수들을 미리 선택해 주는 계수 선택회로를 통해 전건부의 적합도가 0이 되는 모든 규칙들에 대한 연산 회로를 제거하였다. 그리고 귀속도 함수 발생기에서 발생하는 전류 모드 CMOS 회로의 문제점인 팬-아웃(fan-out)수의 제한을 해결하기 위해 전압입력-전류출력의 귀속도 함수발생기를 OTA방식으로 구성하였다. 기존의 방법에 비해 블록 수와 소자수가 큰 폭으로 감소하였음을 확인하였고, 또한 전건부와 언어변수의 수가 증가할수록 더욱 효과적임을 확인하였다.

Abstract

In this paper, we proposed a new circuit construction method that reduced the number of CMOS devices of singleton fuzzy controller(SFC) through the proposing a new membership function circuit(MFC) which uses the language variable selecting and the coefficient selecting circuit. According to the range of input values, we can choose the language variables beforehand which will be used in the inference. So we proposed the new MFC which generates the only necessary language variables. Also, we removed all rules of which adapting degree of their antecedents is zero through proposing the coefficient selecting circuit which beforehand selects the coefficients which will influence the inference result. Though this method, we simplified the structure of SFC and reduced the size of hardware. And to solve the problem in the current mode with respect to the restriction of the fan-out number, voltage-input and current-out membership function circuits are constituted of operational transconductance amplifiers. A membership function circuit which includes the language variable selecting circuit, a minimum operation circuit we implemented by current mode CMOS devices. As a result of applying proposed method, total numbers of blocks and devices were decreased. If the number of variables and antecedents are getting larger, this method is more efficient.

1. 서론

1965년 L.A.Zadeh에 의해 처음 발표된 퍼지이론은 여러 산업장치와 가전제품들에서 널리 이용되었다. 특히 퍼지이론은 제어분야에서 신경망이론 등과 함께 지능제어 이론으로 새로이 부각되어, 제어의 대상이 불분명하거나 모델링이 부정확한 경우, 유용하게 적용될 수 있다.

대부분의 퍼지 정보처리는 기존의 디지털 컴퓨터를 사용하여 소프트웨어적으로 처리될 수 있으나 처리속도, 전력소비, 함수밀도 등에서 최적의 효과를 얻을 수 없었다. 따라서, 퍼지정보처리를 위한 고유의 하드웨어 시스템의 필요성이 부각되었고 이를 개발하려는 연구가 활발히 진행되어 왔다. Yamakawa는 표준 바이폴라 IC기술을 이용하여 전류방식 퍼지논리회로를 구성하였고, 한계차(bounded difference)회로를 기본으로 하여 퍼지논리의 여러 연산 중 일부인 9개의 퍼지논리 기본회로를 구성하여 18개의 핀을 갖는 Semi-Custom IC회로를 제안하였다^[1-2]. Togai등은 측정값이 퍼지집합으로 입력되지 않은 퍼지추론기관의 구조를 제안하였다^[3]. Zhijian등은 전류방식 CMOS IC 기술을 이용하여 9개의 퍼지 연산회로를 구성하고, 모든 규칙은 병렬로 처리되지만 각각의 규칙은 직렬로 처리되는 추론기관의 구조와 회로를 구현하였다^[4]. 또한 Sasaki등은 퍼지추론에서 MIN, Max연산을 이용할 때 유용한 다입력 MIN, MAX회로를 제안하였다^[5]. 한편 실제 시스템에서는 하나의 수치로 표시되는 확정치를 필요로 하는 경우가 많기 때문에 추론결과를 확정치로 변환하여 주는 과정이 필요하다. 이 과정에서 무게중심법을 주로 사용하게 되는 데, 이때 연산에 필요한 제산기는 하드웨어의 규모를 크게 증가시키고 처리속도를 저하시키는 요인이 된다. 이러한 문제점 때문에 제산기를 제거하기 위한 연구들이 진행되어 왔고 그 결과 Sasaki와 Ishizuka등은 제산기를 제거하는데 성공하였다^[6-7]. 기존의 제산기를 제거한 퍼지제어기 구성방법은 한 규칙당 한 개의 추론블록으로 구성된 회로를 제안하였다. 그러나, 퍼지추론시 하나의 전진부에 입력된 한 입력값에 대해 단지 두 개의 언어변수를 나타내는 함수값만 연산에 사용되므로 나머지 언어변수를 나타내는 함수값은 연산에 영향을 주지 않는다. 또한, 전진부의 적합도가 0이 되는 모든 규칙들은 실질적으로 추론의 결과에 전혀 영향을 주지 않게 되어 모든 규칙에 대하여 연산을 수행하는 것은 매우 비 효율적이다^[8]. 또한 기존의 퍼

지제어기의 규칙블록은 언어변수와 전진부의 증가에 따라 v^p (v :언어변수의 개수, p :전진부 개수)만큼 증가하게 되어 하드웨어의 규모가 크게 증가한다.

본 논문에서는 언어변수 선택기를 이용한 새로운 귀속도 함수 발생기를 제안하여 연산에 영향을 주지 않는 귀속도 함수 발생을 제거하도록 하였다. 그리고 추론결과에 영향을 주는 계수들을 선택하는 방법을 제안하여 퍼지추론시 전진부의 적합도가 0이 되는 모든 규칙들에 대한 추론 회로들을 제거하도록 하였다. 귀속도 함수 발생기의 전류모드 CMOS회로의 문제점인 fan-out수의 제한을 해결하기 위해 전압-입력, 전류-출력인 OTA(Operational Transconductance Amplifier)방식의 귀속도 함수 회로를 구성하였다. 제안된 방법을 이용하여 전진부가 2개이고 퍼지변수값이 각각 3개인 경우에 대한 퍼지제어기를 구성하고 T-Norm(Min연산, 한계곱 연산)회로, 계수회로를 전류모드 CMOS로 구현하였다. 또한 언어변수선택기와 계수 선택회로를 구현하고 3변수 귀속도 함수 회로와 함께 범위가 다양한 5변수에 대해서도 추론을 수행할 수 있는 귀속도 함수 회로를 구현하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 퍼지의 기본이론인 퍼지집합, 언어변수, 퍼지연산, 퍼지추론등을 다루고 제3장에서는 싱글톤퍼지제어기의 기본 설계방법과 제산기를 제거하기 위한 T-Norm연산자, 그리고 T-Norm을 이용한 기존의 방법에 대해 다루었다. 제4장에서는 제안된 퍼지제어기의 구성원리와 제안된 방법에 의한 블록별 회로구현에 대한 내용을 다루고, 제5장에서는 기존방법들과의 전체 블록수와 소자수의 비교 및 검토를 하고 제6장에서 본 논문의 결론을 맺었다.

2. 퍼지 이론

2.1 퍼지 집합

현재 광범위하게 사용되고 있는 디지털논리(이진논리)는 그 연산과정의 편리성과 회로구현의 용이성의 장점을 가지고 있으나, "0"과 "1", "Low"와 "High"등의 이진값만을 사용함으로써 이진값 사이에 존재하는 값의 처리가 쉽지 않은 단점을 가지고 있다. 즉, "신호가 있다", "신호가 없다"에 관한 처리는 용이하나 "신호가 조금 있다"라는 식의 애매한 상황에서는 그 처리가 용이하지 못하며 "있다" 또는 "없다"의 어느 한쪽으로 인식함으로써 원래 정보를 왜곡시킬 수 있다. 이러한 이진논리의 부적합성에 대하여 이를 적절

하게 처리하기 위한 새로운 집합이론이 필요하다고 주장되었고 그 새로운 집합이론으로서 퍼지집합이론이 발표되었다.

본 절에서는 퍼지집합의 기본적 용어들을 간단히 정의하고자 한다.

<정의 1>

귀속도(membership grade, membership degree)

집합의 원소가 그 집합에 소속되는 정도를 단위구간 [0,1]사이의 값으로 나타낸 것을 말하며, 전체집합 U의 퍼지부분집합 A의 원소가 그 집합에 속하는 정도를 $\mu_A(x)$ 로 나타낸다.

귀속도 함수(membership function)

퍼지집합의 각 원소들의 귀속도를 표현한 함수.

<정의 2>

퍼지집합(fuzzy set)

집합의 각 원소가 집합에 속하거나 속하지 않는 두 가지중 하나로 결정되지 않고 각각의 귀속도를 취하는 원소들로 구성되는 집합.

<정의 3>

언어변수(linguistic value)

홍길동이라는 사람을 묘사하는 표현에는 키, 체중, 나이, 외모 등의 항목들이 있을 수 있다. 이 항목들 각각에 대하여 키가 크다, 작다, 체중이 무겁다, 나이가 젊다 등의 언어적 표현이 있을 수 있다. 이와같은 언어적 표현을 각각 항목들에 대한 퍼지 변수라 하고, 이들을 포함하는 상위개념인 키, 체중, 나이 등을 언어변수라 한다^[8,9].

2.2 퍼지 연산

퍼지집합에 사용되는 논리연산은 NOT, OR, AND 등이 있으며 다음과 같은 공리를 정의하여 연산을 행한다.

1) 부정연산

$N : [0,1] \rightarrow [0,1]$ 이 다음 조건을 만족할 때, 이를 부정연산이라 한다.

$\forall a, b \in [0, 1]$

- ① $N(0) = 1, N(1)=0$: 경계조건
- ② $a < b \rightarrow N(a) \geq N(b)$: 단조 비증가
- ③ $N(N(a)) = a$: 이중 부정

2) T-norm

보통 집합의 AND와 비교되는 것으로 $T:[0,1] \times [0,1] \rightarrow [0,1]$ 이 다음의 조건을 만족할 때, 이를 T-norm(곱연산)이라 한다.

$\forall a, b \in [0, 1]$

- ① $T(0, 0) = 0, T(a,1)=T(1,a)=a$: 경계조건
- ② $a \leq c, b \leq d \rightarrow T(a, b) \leq T(c, d)$: 단조성
- ③ $T(a, b) = T(b, a)$: 교환 법칙
- ④ $T(T(a, b), c) = T(a, T(b, c))$: 결합 법칙

3) S-norm

보통 집합의 OR와 비교되는 것으로 $S:[0,1] \times [0,1] \rightarrow [0,1]$ 이 다음의 조건을 만족할 때, 이를 S-norm(합연산)이라 한다.

$\forall a, b \in [0, 1]$

- ① $S(1, 1) = 1, S(a,0)=S(0,a)=a$: 경계조건
- ② $a \leq c, b \leq d \rightarrow S(a, b) \leq S(c, d)$: 단조성
- ③ $S(a, b) = S(b, a)$: 교환 법칙
- ④ $S(S(a, b), c) = S(a, S(b, c))$: 결합 법칙

2.3 퍼지 추론

퍼지 실용화의 가장 중요한 기법으로 퍼지 추론(Fuzzy Inference)이 있다. 주어진 규칙과 사실의 모임으로부터 논리적으로 타당한 새로운 사실을 얻어내는 과정을 추론이라 한다. 어떤 규칙과 사실로부터 추론의 간단한 예를 보이면 다음과 같다.

규칙 : 키가 큰 사람은 몸무게가 무겁다.

사실 : 홍길동은 키가 약간 크다.

추론결과 : 홍길동은 몸무게가 약간 무겁다.

위 예와 같이 홍길동의 키가 “약간 크다”와 같은 애매한 표현은 이진논리로는 처리할 수 없는 표현이며 이러한 경우에 퍼지논리를 적용한 퍼지 추론기법을 적용할 수 있다. 현재 가장 기본적인 추론방법으로 전방향 추론(Generalized Modus Ponens)과 후방향추론(Generalized Modus Tollens)이 있다. 이러한 추론 기법의 간단한 예를 보이면 다음과 같다.

1) 전방향 추론(Generalized Modus Ponens : GMP)

(규칙) If X is A then Y is B

(사실) X is A'

(결론) Y is B'

2) 후방향 추론(Generalized Modus Tollens : GMT)

(규칙) If X is A then Y is B

(사실) Y is B'

(결론) X is A'

위 예에서와 같이 If 다음에 나오는 명제(X is A)를 전건부라 하고, then 다음에 나오는 명제(Y is B)를 후건부라 한다. 이와같이 사실에 해당하는 어떤 입력을 받아 미리 구성된 규칙들을 사용하여 그 결과를 출력으로 도출하는 것이 추론을 수행하는 퍼지논리 제어기(Fuzzy Logic Controller : FLC)의 기본적인 방법이다.

3. 싱글톤 퍼지 제어기의 설계

3.1 추론 원리

퍼지제어기 설계방법을 논하기에 앞서 퍼지 제어기의 추론 원리를 정립하여야 하며 필요한 If-then 규칙베이스를 구성하여야 한다. If-then 규칙베이스의 한 예를 다음의 표 3.1에 구성하였다.

표 3.1 If-then 규칙베이스

Table 3.1 If-then Rule Base

규칙1> If x is NE and y is NE, then z is C1.
규칙2> If x is NE and y is ZO, then z is C2.
규칙3> If x is NE and y is PO, then z is C3.
규칙4> If x is ZO and y is NE, then z is C4.
규칙5> If x is ZO and y is ZO, then z is C5.
규칙6> If x is ZO and y is PO, then z is C6.
규칙7> If x is PO and y is NE, then z is C7.
규칙8> If x is PO and y is ZO, then z is C8.
규칙9> If x is PO and y is PO, then z is C9.

여기서, x, y는 각각 입력이며,

NE는 음수 값(Negative), ZO는 0의 값(Zero),

PO는 양수 값(Positive)를 각각 의미한다.

* 총 규칙수 = (언어변수개수)^{입력개수} = 9

표 3.1의 규칙베이스는 2입력 (x, y)로 부터 3변수(NE, ZO, PO)에 의해 만들 수 있는 9개의 규칙, C1~C9을 나열하였다. 그림 3.1은 이 세 퍼지변수에 대한 귀속도 함수의 예이다.

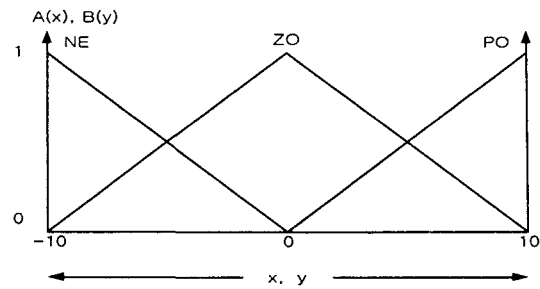


그림 3.1 2입력-3변수에 대한 귀속도 함수의 예
Fig. 3.1 An example of membership functions for 2input-3variable.

3.2 싱글톤 퍼지 제어기

싱글톤 퍼지제어기는 다음과 같은 퍼지규칙에 따라 설계될 수 있다.

If x is $A_1(x)$ and y is $B_1(y)$ then z is C_1

If x is $A_2(x)$ and y is $B_2(y)$ then z is C_2

여기서, x와 y는 각각 입력이며 z는 출력이다.

$A_i(x)$ 와 $B_i(y)$ 는 입력 x와 y에 대한 각각의 귀속도 함수값이며 C_i 는 추론의 결과이다. (i =1, 2)

싱글톤 퍼지제어기는 이러한 퍼지규칙에 따라 설계된 제어기에 입력 x' 와 y' 를 입력하였을 때 출력 z' 에 대한 수식을 다음과 같이 정의하였다.

$$z' = \frac{\sum_i g_i \cdot C_i}{\sum_i g_i} \quad (1)$$

$$g_i = A_i(x') \wedge B_i(y') \quad (2)$$

여기서, \wedge 는 MIN 연산기호를 나타내며, g_i 값은 $A_i(x)$ 와 $B_i(y)$ 에 대한 T-norm연산의 결과값이다.

위 식(1)과 같이 출력을 얻는 과정에서 나눗셈 연산이 사용되므로 회로에 제산기가 필요하게 되는데, 이는 하드웨어의 규모를 크게 증가시키고 처리속도를 저하시키는 요인이 된다. Sasaki는 새로운 T-norm연산자를 제안하여 제산기 제거에 성공하였다.[6] 그는 '임의의 한 입력값에 대한 전건부 귀속도 함수들의 총합은 1(orthogonal condition)이고 서로 중복되는 귀속도 함수들의 수가 2개 이하'라는 조건을 만족하고, T-norm연산자를 MIN연산과 한계곱 연산의 합을 사용하여 출력시 제산기를 제거하였다.

Sasaki가 제안한 새로운 T-norm연산자는 다음의

식 (3)과 같다.

$$x \cap y = \frac{(x \wedge y) + (x \odot y)}{2}$$

$$x \odot y = \begin{cases} (x + y) - 1, & \text{if } x+y \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

이러한 T-norm 연산자를 사용하게 되면 (1)식의 $\sum_i g_i$ 가 1이 되어 출력 z' 에 대한 식은 다음의 (4),(5)와 같이 된다.

$$z' = \sum_i g_i \cdot C_i \quad (4)$$

$$g_i = A_i(x') \cap B_i(y') \quad (5)$$

(본 논문에서는 $\sum_i g_i = 1$ 에 대한 증명은 생략하며 이에 관한 내용은 참고문헌[6]을 참조한다.)

3.3 Sasaki 퍼지제어기의 설계방법과 구조

앞의 3.1과 2절의 내용을 바탕으로 식 (4)의 추론결과를 논리연산으로 나타내면 다음과 같다. 편의상 전건부 x', y' 에 대한 귀속도 함수값 $A_i(x'), B_i(y')$ 를 간단히 A_i 와 B_i 로 표시하기로 한다.

$$z' = \sum_i g_i \cdot C_i$$

$$= (A_1 \cap B_1)C_1 + (A_1 \cap B_2)C_2 + (A_1 \cap B_3)C_3$$

$$+ (A_2 \cap B_1)C_4 + (A_2 \cap B_2)C_5 + (A_2 \cap B_3)C_6$$

$$+ (A_3 \cap B_1)C_7 + (A_3 \cap B_2)C_8 + (A_3 \cap B_3)C_9 \quad (6)$$

입력 x, y 에 대한 귀속도 함수 값 A, B 를 발생시키는 귀속도 함수 발생기(Membership Function Circuit : MFC)를 구성하고 T-Norm(Min, BP연산: \cap)을 수행하는 연산기 회로와 해당하는 규칙의 계수값을 발생시켜 승산 회로를 구성하여 이를 모두 합해주는 구조의 회로로 퍼지제어기를 구성한다.

그림 3.2는 Sasaki에 의해 제안된 9개의 추론블럭을 갖는 퍼지제어기이다.

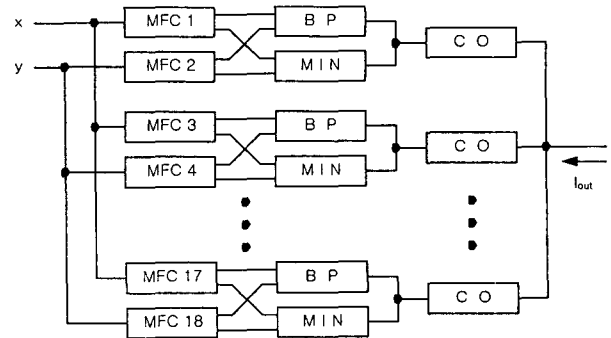


그림 3.2 Sasaki의 퍼지제어기 블럭도 (2입력-3변수)
Fig. 3.2 Block Diagram of Sasaki's fuzzy controller (2inputs-3variables)

4. 제안된 퍼지제어기의 구성원리

식 (4)와 (5)를 통해 설계된 Sasaki의 퍼지제어기에 대하여 본 논문에서는 T-Norm연산자의 성질을 이용하여 언어변수와 계수를 선택함으로써 보다 개선된 회로를 제안하였다.

4.1 언어변수 선택방법과 계수 선택방법

Sasaki에 의해 제안된 방법은 그림 3.2에서 보는 바와 같이 입력 x, y 에 대한 귀속도 함수 발생회로들과 T-Norm연산회로, 계수회로를 규칙 하나당 한 블록씩 구성하여 규칙의 갯수 만큼 모든 블록을 접속하여 추론을 수행하도록 되어있다. 따라서 모든 언어변수(NE, ZO, PO)에 해당하는 함수값을 모두 발생시키도록 회로가 구성되어 있고, 또한 모든 9개의 규칙에 대하여 추론을 수행한다. 그러나 실제추론에 있어서는 3.2 절의 주어진 조건을 만족해야 하므로, 첫째 입력 x, y 가 인가될 때 그 입력값의 범위에 대한 언어변수가 각각 2개씩으로 미리 결정되게 된다. 예를 들어, 그림 3.1에서 임의의 입력 x 가 $[-10, 0]$ 구간에 존재할 때 PO 곡선에서의 귀속도 함수값은 0이 되며, T-Norm연산의 결과 역시 0이 되어 추론에 영향을 주지 않게 된다.

[예제 4.1]

그림 3.1과 같이 2입력-3변수에 대한 귀속도 함수를 갖는 퍼지제어기에 입력(x, y)가 (5, -5)로 주어졌을 때 출력함수를 구하면 다음과 같다.

표 3.1의 If-then 규칙베이스를 사용하여 추론을 하게되면 다음과 같은 결과를 얻을 수 있다. 여기서, $x =$

5, $y=-5$ 이므로 "x is NE"에 해당하는 귀속도 함수 값은 0, "y is NE"에 해당하는 귀속도 함수 값은 0.5가 된다. 즉, $A_1(x=5)=0$, $B_1(y=-5)=0.5$.
나머지, 귀속도 함수 값을 모두 구하면 다음과 같다.

- "x is ZO"에 해당하는 $A_2(x=5)=0.5$
- "x is PO"에 해당하는 $A_3(x=5)=0.5$
- "y is ZO"에 해당하는 $B_2(y=-5)=0.5$
- "y is PO"에 해당하는 $B_3(y=-5)=0$

따라서, 입력 $(x, y)=(5, -5)$ 에 대한 추론 결과 z 는 식 (6)을 통해 다음과 같이 얻을 수 있다.

$$\begin{aligned}
 z' &= \sum_i g_i \cdot C_i \\
 &= (0 \wedge B_1)C_1 + (0 \wedge B_2)C_2 + (0 \wedge B_3)C_3 \\
 &\quad + (0.5 \wedge 0.5)C_4 + (0.5 \wedge 0.5)C_5 + (0.5 \wedge 0)C_6 \\
 &\quad + (0.5 \wedge 0.5)C_7 + (0.5 \wedge 0.5)C_8 + (0.5 \wedge 0)C_9 \\
 &= 0.5C_4 + 0.5C_5 + 0.5C_7 + 0.5C_8
 \end{aligned}$$

이와 같이 임의의 하나의 입력에 대하여 그 출력으로 나타나는 귀속도 값은 항상 2개 뿐이다. 결과적으로 2개의 입력 x, y 에 대하여 각각 2개 씩 4개의 귀속도 함수값만이 출력연산에 영향을 주게 된다. 이는 언어변수의 수가 늘어도 마찬가지로 귀속도 함수의 개수는 단지 모든 전전부의 개수에 두배한 만큼만 갖게 된다. 또한 입력에 대한 귀속도 함수 값 들은 서로 보수 관계가 된다.

한편, 추론에 영향을 미치게 되는 4개의 계수들도 입력의 구간에 의해 미리 결정되게 된다. 앞의 예제 4.1에서 살펴본 바와 같이 입력 x 가 구간 $[0, 10]$ 에 있고, y 가 구간 $[-10, 0]$ 에 있을 때 사용될 4개의 귀속도 함수들은 (A_2, A_3, B_1, B_2) 이다. 규칙베이스에서 이 4가지 언어변수를 사용하는 규칙은 4, 5, 7, 8이므로 결국 추론에 필요한 계수들은 C_4, C_5, C_7, C_8 이 된다. 이러한 방법으로 모든 경우에 대하여 이러한 두가지 성질을 표 4.1과 같이 정리할 수 있다.

표 4.1 입력구간과 언어변수, 계수와의 관계
Table 4.1 Relation among the input range, linguistic variables, and coefficients.

x 구간	y 구간	사용될 귀속도함수값	결정될 계수들
$[-10, 0]$	$[-10, 0]$	A_1, A_2, B_1, B_2	C_1, C_2, C_4, C_5
$[-10, 0]$	$[0, 10]$	A_1, A_2, B_2, B_3	C_2, C_3, C_5, C_6
$[0, 10]$	$[-10, 0]$	A_2, A_3, B_1, B_2	C_4, C_5, C_7, C_8
$[0, 10]$	$[0, 10]$	A_2, A_3, B_2, B_3	C_5, C_6, C_8, C_9

[예제 4.2]

표 4.1을 이용하여 퍼지제어기 입력 (x, y) 가 $(-5, 5)$ 로 입력될 때의 추론결과를 구해보면 다음과 같다. 표 4.1로부터 입력 x 와 y 가 각각 $[-10, 0], [0, 10]$ 구간 내에 존재함을 알 수 있다. 따라서 추론결과 z' 는 다음과 같이 나타난다.

$$\begin{aligned}
 z' &= \sum_i g_i \cdot C_i \\
 &= 0.5C_2 + 0.5C_3 + 0.5C_5 + 0.5C_6
 \end{aligned}$$

Sasaki에 의해 제안된 퍼지추론 식으로 부터 T-Norm 연산의 성질을 이용하여 예제 4.1, 4.2와 같이 4개의 관계식으로 축소할 수 있었다.

4.2 제안된 퍼지제어기의 구조

본 절에서는 앞절에서 논의된 내용을 토대로 그림 4.1와 같이 2변수 3전전부의 퍼지제어기의 구조를 설계하였다.

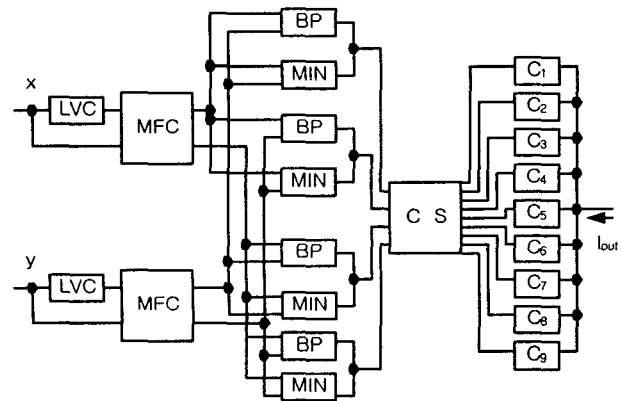


그림 4.1 제안된 퍼지 제어기 블록도
Fig. 4.1 The proposed block diagram of fuzzy controller.

언어변수선택기(Language Variable selecting Circuit)를 나타내는 블록을 LVC로 표시하였고, 계수선택(Coefficient Selecting)블록을 CS로 표시하였다. 설계

된 회로의 입력 x, y 가 입력될 때, LVC에서 입력된 구간을 MFC에 알려주고 그에 따른 귀속도 함수를 MFC에서 발생시켜 준다. 그 후 발생된 4개의 값을 T-Norm(Min, BP(한계값))연산하여 계수 선택블록에서 선택된 계수와 곱하도록 연결해 준다. MFC가 전전부의 갯수만큼만 필요하므로 퍼지변수의 증가와 관계없이 일정하며 연산블록도 기존의 방법이 v^p (v :퍼지변수의 갯수, p : 전전부의 갯수)개가 필요한데 비해 본 논문에서 설계된 회로에서는 단지 2^p 개만 필요하다.

4.3 블록별 회로 구현

본 절에서는 2입력-3변수에 대한 블록별 회로를 구현하였으며 각 블록들은 변수와 전전부가 증가되어도 쉽게 확장가능하다.

1) 언어변수 선택기 회로

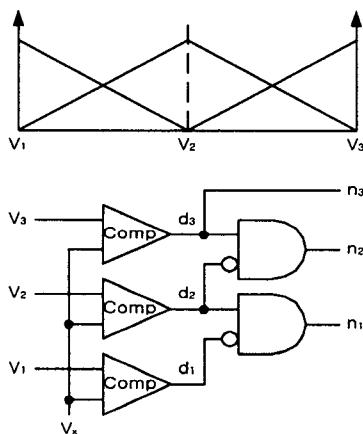


그림 4.2 언어변수 선택기의 제어신호 발생기 구조
Fig. 4.2 The structure of control signal generator of LVC.

표 4.2. d 신호와 n신호표

Table 4.2. The table of d and n signal.

	$V_1 < V_x < V_2$	$V_2 < V_x < V_3$	$V_3 < V_x$
d_3	0	0	1
d_2	0	1	1
d_1	1	1	1

	$V_1 < V_x < V_2$	$V_2 < V_x < V_3$	$V_3 < V_x$
n_3	0	0	1
n_2	0	1	0
n_1	1	0	0

그림 4.2의 회로는 입력 V_1, V_2, V_3 와 전압 V_x 와의 비교를 통해 표 4.2과 같이 d 신호를 발생하게 된다. 이 d신호를 통해 언어변수를 선택하는 스위치들의 ON/OFF 제어신호를 제어신호 n을 발생시켜 사용하게 된다. 이 n신호는 다음절의 계수 선택회로에서도 계수회로로 연결되는 스위치들의 제어신호로 사용하게 된다. d신호를 n신호로 변환시키는 변환회로는 다음의 그림 4.3와 같다.

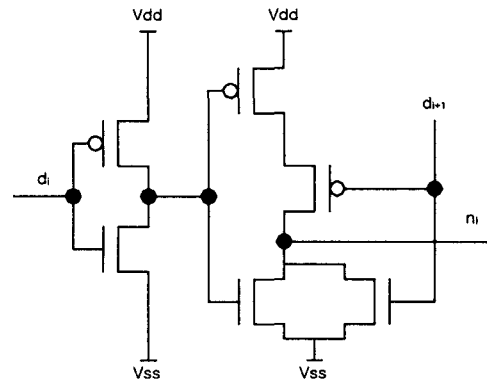


그림 4.3 신호 변환기.
Fig. 4.3 Signal Converter.

그림 4.2와 4.3의 회로를 통해 3변수 귀속도 함수 발생기와 언어변수 선택기를 그림 4.4과 같이 설계하였다. 그림 4.4의 회로에서 왼쪽의 CMOS만으로 구성된 하나의 차동증폭단이 MFC기능을 하고 나머지 회로들이 LVC에 해당된다. LVC에서 발생된 n신호가 스위치 소자를 ON시켜 MFC의 우측에 기준전압을 공급해 주게 된다. 이 V_{cx} 에 의해 언어변수 2개가 결정되어 입력값의 구간에 대한 필요한 귀속도 함수값을 발생시켜 준다. 또한 기준전류 I_{ss} 값이 퍼지값 1을 나타내는 값이 된다.

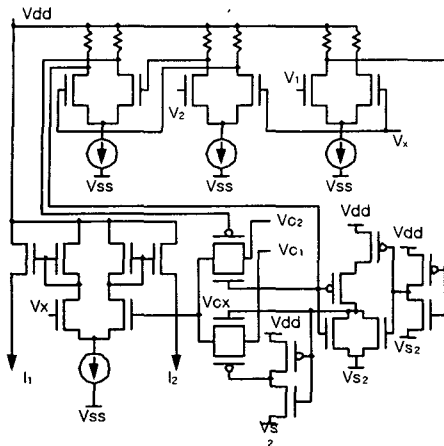


그림 4.4 언어변수 선택기를 연결한 귀속도 함수 회로(3변수)

Fig. 4.4 The membership function circuit with language variable selecting circuit(3-variables).

기존의 방법은 변수가 증가할 때마다 차동증폭단 여러개를 병렬연결하여 구성하였으므로 MFC의 크기가 증가하게 된다. 제안된 방법을 사용할 경우 변수의 갯수가 증가하여도 하나의 차동증폭단만으로 회로를 구성할 수 있어 하드웨어의 크기를 개선할 수 있다.

5. 비교 및 검토

본 논문에서 제안한 방법을 이용하면 전건부의 개수가 p 개로 증가하고 각 전건부의 퍼지변수들이 v 개로 증가할 때, 2^p 개의 규칙에 대한 연산만을 수행하는 회로로 구성된다. 반면 기존의 방법을 이용할 경우, 동일조건에서 v^p 개의 규칙에 대한 연산을 수행하는 회로로 구성되므로 전체적인 블록수가 크게 증가한다.

제안된 방법에서 첨가된 계수 선택회로부분과 언어변수선택회로가 퍼지변수와 전건부의 증가에 따라 다소 복잡해지는 단점을 안고 있다. 그러나 기존의 방법은 퍼지변수의 증가에 따라 모든 블록의 회로가 증가하는 반면, 제안된 방법은 계수 선택회로와 언어변수선택회로만 다소 증가하므로 전체 소자수 비교에서도 크게 감소된 결과를 얻었다. 적절한 비교를 위하여 제어기의 기본적 구조가 같아야 하므로 기존의 Sasaki의 MFC회로와 본 논문의 LVC가 연결된 MFC는 비슷한 소자수를 가지게 되므로 동일한 블록으로 비교하였다. 표 5.1에서 기존의 방법보다 계수 선택회로가 한 블록 더 필요하지만 전체적인 블록수가 크게

감소하며, 또한 표 5.2의 소자수 비교에서도 제안된 방법을 사용할 경우 회로의 전체 소자수가 크게 감소함을 알 수 있다.

표 5.1 블록수 비교표($v = 3, 5, 7, \dots$).

Table 5.1 Comparison table of the number of blocks.

	Sasaki[6]		본 논문	
변수, 전건부개수	3, 2	v, p	3, 2	v, p
MFC	18	pv^p	2	p
Min, BP	9	v^p	4	2^p
Co-selection	.	.	1	1
총 블록수	27	$(p+1)v^p$	7	2^p+p+1

표 5.2 소자수 비교

Table 5.2 Comparison table of the number of MOS devices.

	Sasaki[6]			본 논문		
변수, 전건부개수	3, 2	5, 2	7, 2	3, 2	5, 2	7, 2
MFC	216	1500	5880	60	100	156
Min, BP	81	225	441	36	36	36
Co-selection	.	.	.	16	128	288
총 블록수	297	1725	6321	112	264	480

6. 결론

추론시 실제 사용되는 언어변수와 규칙은 제한되어 있는데, 기존의 방법으로 퍼지제어기를 구성할 경우, 매 추론 수행시마다 모든 언어변수와 규칙에 대한 추론을 수행할 수 있도록 구성되어 있다. 이로 인해 언어변수와 전건부수가 증가할수록 하드웨어의 규모가 크게 증가하는 문제점이 있다. 따라서, 본 논문에서는 언어변수를 선택하여 함수를 발생시켜 주는 귀속도 함수 발생기 회로를 제안하고, 계수 선택방법을 도입하여 추론에 영향을 주지않는 부분들을 제거할 수 있는 새로운 설계방법 및 제어기 구조를 제시하였다. 또한, 귀속도 함수 발생기를 OTA방식으로 구현하여 팬-아웃(fan-out)제한 문제를 해결하고 전류모드 CMOS로 추론회로들을 구현하였다. 본 논문에서 제안한 회로를 통해 전체적인 퍼지제어기의 구조가 간략화 되고, 퍼지제어기의 블록수와 소자수가 크게 감소하

게 된다. 또한 전진부의 수와 언어변수의 수가 많아 질수록 제안된 방법이 더욱 효과적이다.

참고문헌

- [1] T. Yamakawa, T. Miki, F.Ueno "The design and fabrication of the current mode fuzzy logic semi-custom IC in the standard CMOS IC technology," Proc. of 15th IEEE Int. Symp. Multiple-Valued Logic, pp.76-82, 1985.
- [2] T. Yamakawa, T. Miki "The current mode fuzzy logic integrated circuits fabricated by CMOS process," IEEE Trans. on Computer, pp.161-167, 1986.
- [3] M. Togai, S.Chiu, "A fuzzy logic chip and a fuzzy interface accelerator for real-time approximate reasoning," Proc. of 17th IEEE Int. Symp. Multiple-Valued Logic, pp.25-29, 1987.
- [4] L. Zhijian and J. Hong, "A CMOS current-mode high speed fuzzy logic microprocessor for real-time expert system," Proc. of 20th IEEE Int. Symp. Multiple-Valued Logic, pp. 394-400, 1990.
- [5] M. Sasaki, T. Inoue, Y. Shirai and F. Ueno, "Fuzzy multiple-input maximum and minimum circuits in the current mode and their analyses using bounded-difference equations," IEEE Trans. Computer., vol.39, pp.768-774, Jun. 1990.
- [6] M. Sasaki, F. Ueno, "A fuzzy Logic Function Generator(FLUG) implemented with current mode CMOS circuits," Proc. of 21st IEEE Int. Symp. Multiple-Valued Logic, pp.356-362, May 1991.
- [7] Yoshinori Yamamoto, "A Meaningful Infinite-Valued Switching Function," Proc. of 22st IEEE Int. Symp. Multiple-Valued Logic, pp. 415-422, 1992.
- [8] Msaski Togai, Hiroyuki Watanabe, "Expert System on a Chip: An Engine for Real-Time Approximate Reasoning," IEEE Expert, Fall, pp.55-62, 1986.
- [9] Lofti A. Zadeh, "Fuzzy Logic," IEEE Computer Magazine, pp.83-93, April 1988