

Rule 선택 기법을 사용한 Type-1 Fuzzy Logic Controller의 연산 효율성 향상

조정우*, 박귀태*

고려대학교* 지능시스템 연구실. {amaranth_mrn, gtpark}@korea.ac.kr

Enhancement of Computational Efficiency for Type-1 Fuzzy Logic Controller Using Rule Selection Method

Joh, Jung-Woo*, Park, Gwi-Tae*

Korea University* Intelligent System Research Lab. {amaranth_mrn, gtpark}@korea.ac.kr

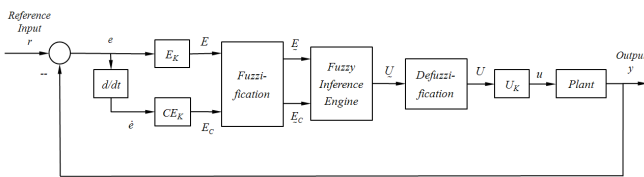
Abstract - 본 논문에서는 제어상황에 따라 Type-1 Fuzzy Logic Controller가 선택적으로 rule을 사용하도록 rule 선택 알고리즘을 제안한다. 그리고 이를 통해 연산 효율성을 높이는 방법에 관해 논한다.

Type-1 Fuzzy Logic Controller는 기존의 제어기에 비해 설계하기 쉽고 성능이 더 뛰어나다. 그러나 제어 변수가 많아질수록 rule의 개수가 늘어나 연산량이 증가하게 된다. 연산량이 많아지면 고성능의 컴퓨터에서는 실시간 연산에 문제가 없으나 산업용 micro controller에서는 실시간 연산을 구현하는데 한계가 발생한다. 본 논문에서는 Type-1 Fuzzy Logic System의 논리구조에 근거하여 Type-1 Fuzzy Logic Controller의 연산량을 감소시킬 수 있는 알고리즘을 제안한다. 제안한 알고리즘은 제어상황에 따라 필요한 rule들만 선택적으로 제어값 도출을 위한 연산에 관여하도록 한다. Matlab 시뮬레이션을 통해 제안한 알고리즘의 유용성과 연산량을 실험하였다. 실험대상은 2륜 이동로봇으로 하였고 step 응답과 전/후진 시 결과를 관찰하였다. 실험 결과 제안한 알고리즘이 기존의 Type-1 Fuzzy Logic Controller에 비해 제어상황에 따라 필요한 rule들만 선택적으로 사용하는 것을 확인하였다. 결과적으로 연산 효율성이 향상되었다.

1. 서 론

기존의 고전적인 제어기법과 비교하여 Type-1 Fuzzy Logic Controller(이하 FLC)는 3가지 중요한 이점들을 가지고 있다. [1]

- ① FLC의 설계과정이 비슷한 제어 성능을 보이는 기존의 물리 모델에 근거한 제어기 설계과정 보다 훨씬 쉽다.
 - ② FLC가 PID 제어보다 더 견고한 제어 성능을 보인다. 이는 FLC가 PID 방법보다 더 넓은 범위의 제어영역을 다룰 수 있기 때문이다.
 - ③ FLC의 변수들을 변화시켜 가면서 다양하고 구체적인 제어 성능을 얻을 수 있다. 이는 FLC의 rule이 사람의 언어로 구성되어 있고 전문가의 지식에 근거하므로 이해하기 쉽고 수정하기 쉽기 때문이다.
- Fuzzy Control System의 구성도를 그림 1에 나타내었다.



<그림 1> Block diagram of Fuzzy Control System

이러한 강력한 장점을 가지고 있는 FLC의 설계방법은 Type-1 Fuzzy Logic System(이하 FLS)의 설계방법에 근거한다. 이 때 기존의 FLS이 가지고 있는 특징들이 FLC에도 나타난다. FLS의 가장 큰 특징은 제어 변수가 늘어남에 따라 rule의 개수도 증가한다는 것이다. [2] Rule의 개수 증가는 연산량의 증가와 직결된다. 고성능 컴퓨터에서는 설계한 FLC의 rule 증가에 따른 연산량 증가가 큰 문제가 되지 않는다. 반면 산업용 MCU에서는 rule 증가에 따른 연산량 증가가 제어의 실시간성을 저해할 수 있다.

FLS의 연산량 증가를 막기 위해 다양한 rule 최적화 기법들이 논문을 통해 발표되었다. 그 중 대표적인 방법이 각각의 rule을 평가한 뒤 평가값이 작은 rule은 FLS에 기여도가 낮다고 보고 가지치기 하는 것이다. 평가방법으로 orthogonal least square method, eigenvalue decomposition method, direct singular value decomposition method, singular value decomposition-QR method [2] 등이 있다. 이러한 rule 가지치기 접근법의 문제점은 다음과 같다. FLC에 기여도가 낮은 rule이라 하더라도 변화가 많은 외부환경과 예상치 못한 돌발상황에 적절히 대처하기 위해서는 다양한 경우를 고려한 rule을 가지고 있어야 한다. 기여도가

낮은 rule을 제거할 경우 이러한 돌발상황에 대처할 수 있는 능력이 없어진다고 볼 수 있다.

본 논문에서는 가지치기식 rule 최적화와는 다른 방법으로 연산량을 감소시킬 수 있는 알고리즘에 대해 논한다. 고안한 알고리즘의 핵심은 다음과 같다. 적절한 개수로 rule이 구성되면 더 이상 rule의 개수를 줄이지 않는다. FLS의 논리구조에 근거하여 각각의 제어상황에 따라 필요한 rule들만 선택적으로 제어값 연산에 관여하도록 한다. 이러한 방법의 이점은 제어상황에 따라 불필요한 rule들은 연산에서 제외하므로 연산 효율성이 높다는 것이다. 본 논문의 구성은 다음과 같다. 2. 본론에서 FLS에 근거한 기존의 FLC의 연산과정을 기술한다. 이어 제안하는 FLC 알고리즘을 정의한다. 3. 실험 및 결과분석에서 각각의 제어기에 대해 2륜 이동로봇을 대상으로 Matlab 시뮬레이터로 실험하고 결과를 비교 분석한다. 4. 결론에서 실험결과에 근거하여 본 논문에서 제안하는 Rule Selection Method Based FLC 알고리즘의 유용성에 대해 결론짓는다.

2. 본 론

2.1 Type-1 Fuzzy Logic Controller

FLS에 근거하여 기존의 FLC가 입력된 변수들을 퍼지화하고 rule base와 추론하여 결과값을 도출하는 과정은 다음과 같다.

2.1.1 Fuzzification

FLC에 변수가 입력되면 퍼지화를 하게된다. Singleton과 non-singleton 퍼지화 기법이 있고 후자의 경우 다양한 모양으로 퍼지화가 가능하다. 퍼지화된 입력변수를 다음과 같이 표현한다.

$$\mu_{x_i}(x_i) \quad (1)$$

여기서 '1'은 변수를 구별하기 위해 표기한 것이다. x_i 는 첫 번째 입력 변수가 가지는 값의 전체영역을 뜻하고 x_i 는 그 전체영역에서 하나의 특정 입력값을 나타낸다. μ 는 x_i 에서의 membership grade를 뜻한다.

2.1.2 Rule Base

추론을 하기 위해서는 rule을 정해야 한다. Rule들은 전문가의 지식으로부터 얻을 수 있고 다음과 같이 IF-THEN 형태로 구성한다.

$$R^l: IF x_1 \text{ is } F_1^l \text{ AND } \dots \text{ AND } x_p \text{ is } F_p^l, \text{ THEN } y \text{ is } G^l, l=1, \dots, M \quad (2)$$

l 은 l 번째 rule을 의미한다. 여기서는 전체 M 개의 rule이 있다고 가정한다. 입력의 개수는 p 개이고 각각의 입력에 대해 Membership Function(이하 MF)이 할당되어 있다.

2.1.3 Inference Engine

퍼지화된 입력값과 rule이 구성되면 이들로부터 추론을 통해 결과값을 얻을 수 있다. 입력값은 퍼지화되어 있으나 rule은 최종적인 membership grade를 모르므로 추론 전에 각각의 rule에 대한 membership grade를 구한다. Rule의 AND는 cartesian product(\times)로 나타낼 수 있고 MF 연산을 위해 t-norm으로 나타낸다. 여기서 t-norm은 minimum 연산으로 정의한다. IF-THEN 관계는 Mamdani implication(\rightarrow) 방법을 사용하고 MF 연산을 위해 minimum t-norm으로 정의한다.

수식 (3)에서 \star 는 minimum t-norm을 나타낸다. 퍼지화된 입력값과 membership grade를 구한 l 번째 rule을 사용하여 추론연산을 한다. 여

$$\begin{aligned} R^l: F_1^l \times \dots \times F_p^l &\rightarrow G^l, l=1, \dots, M \\ \mu_{R^l}(x,y) &= \mu_{F_1^l \times \dots \times F_p^l \rightarrow G^l}(x,y) \\ &= \mu_{F_1^l \times \dots \times F_p^l}(x) \star \mu_{G^l}(y) \\ &= \mu_{F_1^l}(x_1) \star \dots \star \mu_{F_p^l}(x_p) \star \mu_{G^l}(y) \end{aligned} \quad (3)$$

기서는 추론연산에 max-min composition 방법을 사용한다.

$$\begin{aligned} \mu_{y'}(y) &= \mu_{X \circ R'}(y) = \sup_{x \in X} [\mu_X(x) \star \mu_{R'}(x, y)] \\ &= \sup_{x \in X} \left[\begin{array}{l} \mu_{X_1}(x_1) \star \dots \star \mu_{X_p}(x_p) \\ \star \mu_{F_1}(x_1) \star \dots \star \mu_{F_p}(x_p) \\ \star \mu_{G'}(y) \end{array} \right] \\ &= \mu_{G'}(y) \star \left[\begin{array}{l} \sup_{x_1 \in X_1} (\mu_{X_1}(x_1) \star \mu_{F_1}(x_1)) \\ \star \dots \star \sup_{x_p \in X_p} (\mu_{X_p}(x_p) \star \mu_{F_p}(x_p)) \end{array} \right] \end{aligned} \quad (4)$$

1개의 rule에 대한 추론연산 결과는 위의 수식 (4)와 같다. 여기서 \circ 은 max-min composition 연산을 뜻한다. M 개의 rule에 대해 모두 수식 (4)의 연산을 하고 전체 rule을 max 연산 하면 최종적인 추론 결과값과 그에 따른 membership grade를 얻을 수 있다.

$$\mu_{y'}(y) = \bigcup_{i=1}^M \mu_{y'_i}(y) \quad (5)$$

2.1.4 Defuzzification

추론결과 얻어진 결과값은 membership grade를 가지는 퍼지화된 형태이다. 제어에 사용하기 위해서는 1개의 값으로 변환해야 한다. 다양한 비퍼지화 방법이 있고 그 중 하나인 무게중심법(centroid)을 수식 (6)에 나타냈다.

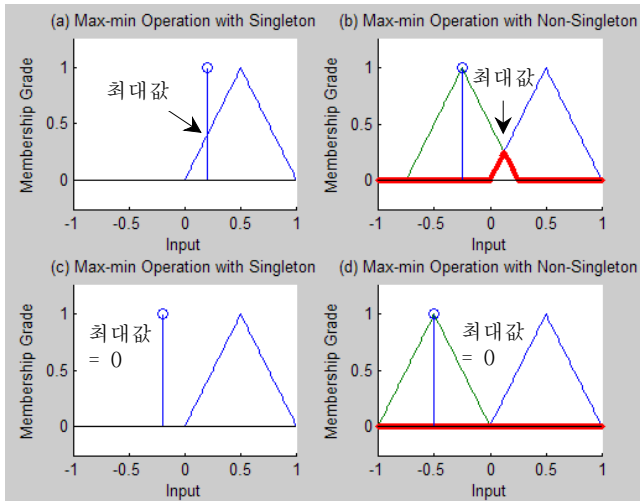
$$y(x) = \frac{\sum_{i=1}^N y_i \mu_{y'}(y_i)}{\sum_{i=1}^N \mu_{y'}(y_i)} \quad (6)$$

2.2 Rule Selection Method Based FLC

수식 (4)에서 1개의 변수에 대한 max-min 연산만 따로 표현하면 수식 (7)과 같다.

$$\sup_{x_1 \in X_1} (\mu_{X_1}(x_1) \star \mu_{F_1}(x_1)) \quad (7)$$

수식 (7)을 삼각형의 MF를 사용하여 연산하는 예를 그림 2에 나타냈다.



<그림 2> 삼각형 MF과 다양한 입력값에 대한 max-min 연산결과

여기서 singleton/non-singleton 퍼지화된 다양한 입력값에 대해 경우에 따라 max-min 연산결과가 0보다 큰 값을 가질 때도 있고 0인 경우도 있다는 것을 확인할 수 있다. 즉 수식 (4)에서 p 개의 max-min 연산 중 1개라도 연산결과가 0이 되면 max-min 연산 간 minimum t-norm에 의해 나머지 변수의 max-min 연산결과와 무관하게 수식 (4)는 최종적으로 0의 값을 가지게 된다. 그러므로 정의되어 있는 rule MF $\mu_{F_i}(x)$ 의 위치와 퍼지화된 입력값 $\mu_X(x)$ 의 범위를 추론연산 전에 미리 확인하여 입력변수들 중 1개라도 max-min 연산결과가 0이 되는 위치로 입력값과 rule MF이 형성되면 그 rule은 추론연산 시 고려하지 않아도 된다. 이러

한 알고리즘을 의사코드(pseudo-code)로 표현하면 다음과 같다.

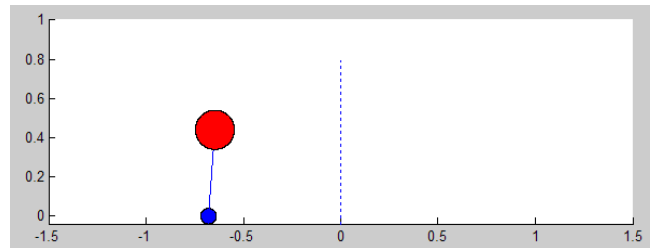
```

index[6] = [0 0 0 0 0 0] /* 3개의 MF에 대한 인덱스 */
i_output = 1;

if (input < a)
    index[0] = 1;
if (input >= a && input < b)
{
    index[1] = 1;
    index[2] = 1;
}
.
.
.
for (i=1 : i<sizeof(index) : i++)
{
    if (index(i) == TRUE)
    {
        Output(i_output) = Min(Firing level, Consequence MF);
        i_output++;
    }
}
    
```

3. 실험 및 결과분석

제한한 알고리즘을 사용하여 Matlab 상에 가상의 물리공간을 형성하고 여기서 2륜 이동로봇을 제어하는 실험을 하였다. 입력에 대한 퍼지화는 singleton으로 하였다. 실험에 대한 평가는 step 응답과 전/후진 시 사용되는 firing rule의 개수를 동일한 시간동안 누적하여 확인하는 방식으로 하였다. 실험에 사용한 Matlab 시뮬레이터를 그림 3에 나타냈다.



<그림 3> 2륜 이동로봇 Matlab 시뮬레이터

<표 1> FLC와 RSM-FLC의 제어상황에 따른 rule 누적 사용개수

	FLC	RSM-FLC	사용 Rule 감소율
Step 응답	4887	2130	56.4 %
전 / 후진	5796	2322	59.9 %

표 1의 결과로부터 본 논문에서 제안한 알고리즘을 사용하면 기존의 FLC와 동일한 성능을 유지하면서 제어상황에 따라 rule의 사용개수가 감소한다는 것을 알 수 있다. 사용되는 rule의 개수가 감소한 것은 그만큼 제어에 필요한 연산량이 감소한 것으로 해석할 수 있다.

4. 결 론

FLC의 제어변수가 증가하면 연산량이 증가한다. 연산량을 줄이기 위해 이론적 관점에서 다양한 해결책이 제시되었다.

본 논문에서 제안한 알고리즘을 적용하여 Rule Selection Method Based FLC를 설계하면 기존의 FLC에 비해 firing rule 개수를 줄일 수 있고 연산량을 최소화할 수 있다. 결과적으로 제어변수 증가에 효과적으로 대처할 수 있다.

[참 고 문 헌]

[1] Huaguang Zhang, Derong Liu, "Fuzzy Modeling and Fuzzy Control", Birkhäuser Boston, 2006.
 [2] Jerry M. Mendel, "Uncertain Rule-Based Fuzzy Logic Systems", Prentice Hall PTR Upper Saddle River, NJ, 2001.