퍼지 후건부의 고속 정수연산

채상원* 이상구
한남대학교 컴퓨터공학과
{csw3182*, sglee}@hannam.ac.kr

High-speed Integer Operations in the Fuzzy Consequent Part

Sangwon Chae* Sanggu Lee
Computer Engineering, Hannam University

요 약

지능 시스템에 사용되는 퍼지 데이터를 고속으로 처리하기 위한 퍼지 제어시스템의 중요한 문제점들 중의 하나는 퍼지 추론 및 비퍼지화 단계에서의 수행속도의 개선이다. 특히 후건부의 계산 및 비퍼지화 단계에서의 고속 연산이 더욱 중요하다. 따라서 본 논문에서는 퍼지 제어기의 속도향상을 위해 후건부 단계에서 [0,1]의 실수 연산을 하지 않고, 퍼지 소속함수의 값을 정수형 격자 (400×30)에 매핑시켜 고속의 정수 덧셈 연산만으로 수행할 수 있는 알고리듬을 제안한다.

1. 서 론

퍼지 이론은 과거에는 퍼지 제어를 중심으로 시스템 제어, 공학분야에 국한되어 활용되어 왔으나 현재에는 데이터 마이닝, 전문가시스템, 패턴인식등 새로운 응용분야에서도 퍼지 이론의 효용성을 보이고 있으며 퍼지 이론을 기반으로 설계되는 퍼지 시스템은 인간의 언어적 개념을 정량적 수치로 표시할 수 있다는 장점 때문에 지능 시스템 및 소프트 컴퓨팅 등의 공학적 기술로서 널리 응용되고 있다[1].

퍼지 논리는 특정 집합 A에 대하여 구성원소의 소속 정도를 0과 1사이의 실수로 나타낸다. 이러한 퍼지값은 퍼지 추론과 비퍼지화 과정을 거쳐 시스템에서 요구하는 제어값이 된다. 퍼지 추론은 외부에서 입력되는 조건부의 퍼지 정보에 대해 각각의 소속함수를 통해 소속정도를 구하고 이 소속정도들에 퍼지 제어규칙을 적용하여 적합도를 구한다. 개개의 제어규칙에서 얻어진 추론결과들에 대한 비퍼지화를 통해 제어값을 구하게 된다. 입력되는 퍼지 정보의 소속 정도는 정수값이 아닌 0과 1사이의 실수로 표현되기 때문에 이러한 추론 과정을 거치는 동안 퍼지 제어기는 많은 양의 실수연산을 필요로 한다[2]. 프로세서의 속도에 따라 조금씩 영향을 받지만 정수 연산이 실수 연산에 비해 곱셈의 경우 10배 이상, 나눗셈의 경우 5배정도 빠른 계산 속도를 갖는다. 계산이 복잡해지고 계산량이 많아질수록 이러한 계산 속도의 차이는 더욱 커지게 된다.

일반적으로 퍼지 시스템에서는 전건부에서 각각의 퍼지 규칙에 대한 α값(degree of fulfillment)을 계산하는데에는 그리 많은 계산을 필요로 하지 않는다. 그러나, 후건부의 계산에서는 무게중심(center of gravity)을 계산하는 비퍼지화(defuzzification) 단계에 필요한 많은 양의 실수 연산을 필요로 하게 된다. 실제로 무게중심을 구하는 방법은 적분을 하여야 하므로 많은 시간이 소요되고, 보통의 경우는

$$COG = \frac{\sum x \cdot f(x)}{\sum f(x)} \quad \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

식 (1)과 같이 후건부를 많은 수의 일정한 간격으로 나누어 양자화 시켜 덧셈을 계산한다. 이러한 많은 수의 양자화 레벨을 실수로 계산하기 위해서는 후건부에서 x축(discourse of universe)을 일정한 간격으로 나누어 많은 수의 실수연산을 하여야 한다[3].

따라서 본 논문에서는 후건부에서 많은 양의 실수 연산으로 인한 퍼지 연산의 속도 저하 문제를 근본적으로 해결하기 위해 후건부의 퍼지 소속함수 그래프를 정수형 격자에 매핑하여 정수 덧셈연산만으로 다음의 정수의 격자의 좌표를 계산할 수 있는 새로운 알고리듬을 제안하여 후건부에서 고속으로 정수 연산만을 수행하는 알고리듬을 제안하려고 한다. 따라서 본 논문에서 제안하는 방법은 기존의 실수형 연산에 비해 훨씬 좋은 성능을 얻을 수 있다.

2. 후건부 단계에서 정수형 연산 알고리듬

퍼지 소속함수의 그래프는 대부분 삼각형 또는 사다리꼴의 형태를 취하고 있다. 이러한 함수의 형태에서 퍼지 입력이 단일 값으로 들어올 때, 각 규칙마다 α값을 구하는 것은 그리 많은 양의 계산을 필요로 하지 않지만 후건부의 연산에서는 각각의 퍼지 규칙에 대해 모든 x축 값에 대한 y축의 실수값을 연산해야 하므로 비교적 많은 양의 실수의 연산을 필요로 한다. 또한 비퍼지화 단계에서 무게중심을 구하는 단계에서는 많은 양의 실수의 곱셈 및 나눗셈의 연산이 필요하게 된다[4].

본 본문에서는 정수형 매핑 알고리듬을 사용하여 [0,1]의 실수형의 소속 함수를 가장 가까운 정수형 격자에 매핑시켜 정수값 만을 갖도록 한다. 후건부의 모양을 계산할 때 후건부의 삼각형 또는 사다리꼴의 모양의 각 정점을 잇는 직선 위에 위치한 정수형 격자점들을 연결하여 표현할 수 있다. 이 과정을 그림 1에 나타낸다.

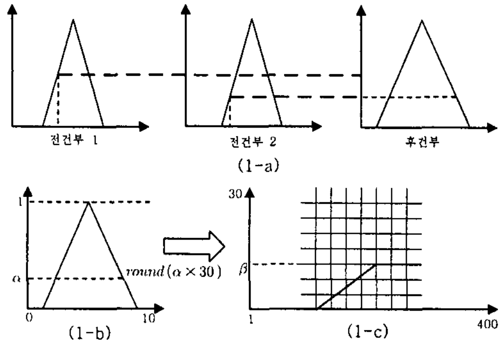

그림 1. 제안하는 퍼지 시스템.
Fig. 1. The proposed fuzzy system.

각각의 규칙에 대해 전건부에서의 적합도(α값, degree of fulfillment)가 계산되면, 후건부에서는 삼각형 또는 사다리꼴의 각 점을 계산하기 위해 가로 400, 세로 30개를 갖는 정수 격자 좌표에 대응시킨다. 전건부에서 α값(실수)이 구해지면 $round(\alpha \times 30)$을 계산한다. 이 값을 β(정수)라 하면, β값이 후건부의 적합도로서 입력된다. 이 때, 그림 1-c에서 후건부의 그래프에 대응되는 격자점들을 계산할 때 정수 덧셈만으로 계산함으로서 매우 빠르게 계산할 수 있다.

2.1 후건부에서의 정수 매핑 알고리듬

후건부에서 추론시 사용되는 그래프의 형태는 여러 개의 직선으로 이루어진 다각형 형태를 가지고 있다. 여기에서 각각의 직선을 표현하는 직선의 방정식을 사용한다면 각 직선의 기울기에 따라 각 점을 계산하는 것은 많은 실수의 곱셈과 덧셈을 사용하게 된다.

그러나, 본 논문에서 제안하는 방법은 실수의 곱셈이 아닌 정수형 덧셈만을 사용하여 소속함수 그래프의 값들을 얻어내므로 기존의 방법에 비해 빠른 연산을 할 수 있다. 중앙점(midpoint)들의 개념을 사용하여 덧셈연산만으로 다음 x격자 좌표의 y값(정수)들을 효율적으로 연산하는 구체적인 알고리듬은 다음과 같다[5].

```
procedure left_line
dx ← x2-x1; dy ← y2-y1; d ← 2dy-dx
xa ← x1; ya ← y1; a ← β
defuzz(xa) ← ya
while ya < a+1 do
   begin
      xa ← xa+1
      begin
         if (d < 0); d ← d+2dy
         else; ya ← ya+1;d ← d+2(dy-dx)
      end
      defuzz(xa) ← ya
   end.
```

그림 2. 정수형 매핑 알고리듬.
Fig. 2. integer mapping algorithm.

따라서 제안된 알고리듬을 사용하면 연속된 격자점들의 y좌표를 구할 때 덧셈연산만으로 가능하다. 물론 제안된 방법에서는 격자좌표를 정수만 사용하므로 실제의 실수값과 약간의 오차가 있다. 그러나 본 논문에서는 x축 400개, y축 30개의 정수 격자를 사용하기 때문에 3장에서 오차분석을 수행하도록 하겠다.

후건부에서의 복잡한 다각형 형태를 그래프로 나타내기 위해서는 기울기의 절대값이 1보다 작은 임의의 직선들을 나타낼 수 있어야 한다. 본 논문에서 제안하는 후건부에서의 정수 격자의 형태는 x축 400개, y축 30개를 사용함으로, 실제 퍼지 추론에 관련되는 모든 후건부의 소속함수의 기울기의 절대값이 1보다 작은 임의의 직선들을 표현할 수 있다. 그림 3에서의 예처럼 삼각형 형태의 그래프에서 α값에 따라 표현되는 후건부의 사다리꼴 모양의 그래프를 표현하기 위해 직선(b), (c)의 매핑 방법에 대해서도 알아본다.

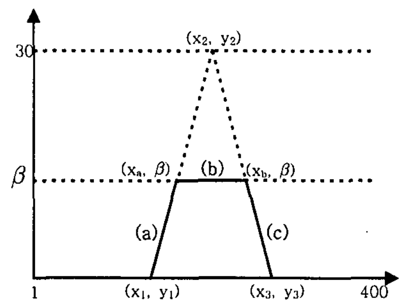

그림 3. β값에 따른 후건부 표현의 예.
Fig. 3. Example of representation of the consequent part using β.

실제로 비퍼지화에서 필요한 부분은 사다리꼴 형태의 그래프이기 때문에 y축에서 정수 $\beta+1$부터 30까지의 부분은 매핑될 필요가 없다. 따라서 정수형 격자점을 표현하는데 있어 β값의 위치까지만 나타내야 한다. 그림 3과 같은 형태의 그래프를 표현할 때에는 순서가 (a)→(c)→(b) 의 순서로 이루어진다. 이것은 (a)→(b)→(c)의 순서로 매핑 된다면 (b)부분을 x축의 어느 지점까지 매핑해야 한다는 것을 판단하기 위해서는 두 점 (x_2, y_2)와 (x_3, y_3)의 직선의 방정식을 이용해서 β값에 대응되는 x값을 계산해야 하는 번거로움이 생긴다. 하지만 (a)→(c)→(b) 순서로 계산을 하게 된다면 (x_3, y_3)에서부터 (x_b, β)까지 정수 격자점에 매핑되기 때문에 직선(c)를 그리게 되면 (b)부분의 직선을 그리는 것은 간단해진다. 여기서 (c)부분을 그리는 방법은 시작점 (x_2, y_2)와 끝점 (x_3, y_3)를 그리는 것이 아니라 (x_3, y_3)에서부터 (x_2, y_2)까지 매핑된다. 이 때 사용되는 알고리듬은 다음과 같다.

```
procedure right_line
dx ← x2-x3; dy ← y2-y3; d ← -2dy-dx
xb ← x3; yb ← y3; a ← β
defuzz(xb) ← yb
while yb < a+1 do
begin
   xb ← xb-1
   begin
      if (d < 0); d ← d-2(dy+dx); yb ← yb+1
      else; d ← d-2dy
   end
      defuzz(xb) ← yb
end.
```

그림 4. 정수형 매핑 알고리듬.
Fig. 4. integer mapping algorithm.

또한 직선(b)는 (a)와 (c)의 결과를 가지고 다음의 그림 5와 같이 쓸 수 있다.

```
procedure middle_line
begin
   a ← β
   x ← xa
   while x < xb do
      begin
         x ← x+1
         defuzz(x) ← a
      end
end.
```

그림 5. 정수형 매핑 알고리듬.
Fig. 5. integer mapping algorithm.

3. 정수형 연산과 실수형 연산의 오차분석

본 논문에서는 실수의 덧셈, 곱셈 연산을 대신할 수 있는 정수의 덧셈 연산만으로 퍼지 추론 과정을 연산할 수 있는 기법을 제안하여 기존의 퍼지 논리 시스템에 비해 고속의 연산 결과를 얻을 수 있다. 그러나 추론 결과의 정확성 측면에서는 [0, 1]의 실수 연산을 사용한 경우와 비교하여 다소 오차가 발생하지만, 무시할 수 있을 만큼 작다. 따라서 본 장에서는 간단한 퍼지 시스템인 에어컨의 모터를 제어하는 시스템에서 기존의 수학적 계산방법과 비교하여 퍼지 연산의 제어값을 비교한다. 대상이 되는 퍼지 제어 시스템은 [3]에 있는 퍼지 시스템으로 하여 9개의 퍼지 규칙과 각 소속함수를 사용한다. 온도가 17℃이고 습도가 32%인 경우 정확한 적분에 의한 식의 경우에 COG 방법에 의한 무게중심은 24.730이다[3]. 이 경우 본 논문에서 제안한 방법과 기존의 실수 연산의 방법을 x축의 구간을 400개와 800개로 나누어 무게중심 값을 비교하여 표 1에 요약하였다.

표 1. 무게중심 비교.
Table 1. Comparison of COGs.

구 분	(400×30) 격자	(800×30) 격자
	무게중심	무게중심
이론치 (적분)	24.73	24.73
기존의 방법[3]	24.77	24.75
제안된 방법	24.83	24.80

(조건 : 온도 = 17℃, 습도 = 32% 일 때)

표 1에서 알 수 있듯이 (400×30) 격자의 경우에 무게중심 값은 기존의 방법인 경우 24.77, 제안된 방법인 경우에 24.83으로 나타났다. (800×30) 격자의 경우에는 무게중심 값은 기존의 방법인 경우 24.75, 제안된 방법의 경우 24.80으로 나타났다. 따라서 오차는 (400×30) 격자 일 때 0.1, (800×30) 격자 일 때 0.07로 줄었다. 본 논문에서 제안된 방법에서 오차를 더욱 줄이려면 (400×30) 격자 대신에 (800×30) 격자와 같이 x축에서의 격자수를 2배로 늘리면 오차는 더욱 더 줄어든다.

4. 결 론

지금까지 대용량의 퍼지 데이터를 처리하기 위해 여러가지 방법들의 고속 퍼지 하드웨어 모듈 및 병렬화 방법에 관한 논문들이 제안되어 왔다. 그러나, 기존의 대부분의 고속 퍼지 제어기들은 고속화 방법에도 불구하고, 전건부 및 후건부의 처리, 무게중심을 구하는 단계에서 [0, 1]의 실수 연산을 통해 퍼지 추론 및 비퍼지 연산을 수행하므로 추론결과를 얻기까지 많은 시간을 실수 연산에 허비하는 문제점을 안고 있다. 일반적으로 퍼지 연산은 전건부보다 후건부 및 무게중심을 구하는 단계에서 대부분의 계산시간을 필요로 하기 때문에 후건부 및 무게중심을 고속으로 처리할 수 있는 하드웨어 모듈 및 알고리듬이 필요하다.

본 논문에서는 후건부의 연산 및 무게중심의 연산에서 후건부를 정수형 격자에 대응시켜 정수의 덧셈연산만으로 고속으로 처리할 수 있는 알고리듬을 제안하였다. 제안한 방법을 지능 시스템이나 고속의 추론을 요하는 퍼지 제어 시스템에 적용하면 높은 성능향상을 얻을 수 있을 것이다. 또한 인공위성으로부터 수집된 원격탐사화상과 같은 대용량의 퍼지 데이터에 대한 실시간 고속처리를 요구하는 시스템에도 잘 활용될 수 있다.

향후의 연구과제로서는 제안된 방법을 이용하여 실제적인 하드웨어 모듈 구현과 구현된 모듈을 이용하여 실질적인 예제를 수행하여 그에 따른 기존의 방법과 제안된 방법의 연산속도 차이를 분석해야 한다.

참 고 문 헌

[1] J. Yen and R. Langari, *Fuzzy Logic : Intelligence*, Prentice Hall, 1999.

[2] E. Cox, *Fuzzy System Handbook*, AP Professional, 1994.

[3] Saade, J. J., "Defuzzification Techniques for Fuzzy Controllers," *IEEE Trans. System, man, and Cybernetics*, vol. 30, no. 1, pp. 223-228, 2000.

[4] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation method," *IEEE Trans. System, man, and Cybernetics-Part B*, vol. 29, no. 1, 1999.

[5] F. S. Hill, *Computer Graphics, 2nd ed*, Prentice Hall, 2002.