퍼지 논리 제어기를 위한 최적의 COA 비퍼지화기

An Optimal COA Defuzzifier for A Fuzzy Logic Controller

조 인 현*, 이 동 석*, 김 종 훈*, 김 대 진*
In-Hyun Cho*, Dong-Seog Lee*, Jong-Hoon Kim*, Daijin Kim*

※이 논문은 1996년도 정보통신부 과학 기초 연구지원 사업 연구비에 의해 연구되었음.

요 약

본 논문은 퍼지 논리 제어기의 제어 성능을 향상시키는 최적의 COA(Center Of Area) 비퍼지화 방법을 제안한다. 제안한 비퍼지화 방법은 플랜트의 제어를 위해 필요한 확정치(Crisp Value)를 얻기 위해 소속 함수값과 소속 함수의 폭을 동시에 이용한다. 주어진 문제에 가장 최적인 소속 함수의 폭은 유전자 알고리즘에 의해 자동적으로 결정된다. 제안한 비퍼지화 방법을 트럭 후진(Truck backer-upper) 제어 문제에 적용하여 얻어진 시뮬레이션 결과로부터 소속 함수만을 고려한 기존의 COA 비퍼지화 방법보다 평균 주행거리 면에서 20% 이상 짧아짐을 확인하였다.

ABSTRACT

This paper proposes an optimal COA(Center Of Area) defuzzification method that improves the control performance of a fuzzy logic controller. The defuzzification method incorporates both the membership values and the effective span of membership functions in calculating a crisp value. An optimal effective span is determined automatically by the genetic algorithm through the training of some typical examples. Simulation of the proposed COA defuzzifier to the truck backer-upper control problem is presented and the control performance of the proposed COA defuzzifier outperforms that of the conventional COA defuzzifier by more than 20% in terms of average tracing distance.

I. 서 론

퍼지 논리 제어기(Fuzzy Logic Controller, FLC)는 퍼지 집합 이론의 한 응용 분야로 최근 공정 제어나 가전 및 산업 전자 분야에서 매우 활발히 연구되고 있다. 특히 시스템의 특성이 복잡하며 입·출력 관계의 정량적 분석이 어렵거나 얻어지는 정보가 정성적이고 부정확한 경우에 기존의 제어기들에 비해 우수한 결과를 나타낸다[1].

그림 1은 퍼지 제어기의 일반적 구성도를 나타낸 것으로 크게 4가지 구성 요소-퍼지화부(Fuzzifier), 추론

*동아대학교 컴퓨터공학과
Dept. of Computer Eng., Dong-A Univ.

엔진부(Inference Engine), 퍼지 규칙 베이스부(Fuzzy Rule Base), 그리고 비퍼지화부(Defuzzifier)로 나뉜다. 각 부분의 동작 설명은 다음과 같다.

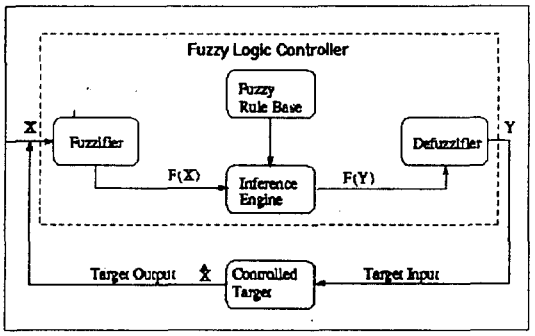


그림 1. 퍼지 제어기의 일반적 구성
Fig. 1 A typical organization of fuzzy logic controller.

퍼지화부에서는 입력 변수의 값을 측정하고, 입력 변수의 영역을 전체 집합범위에 맞도록 크기 변환한 뒤 입력값을 적절한 언어적인 값으로 변환시키고, 추론부에서는 퍼지 관계와 퍼지 논리의 추론 규칙을 사용하여 퍼지 제어 출력을 결정하며, 퍼지 규칙 베이스부에서는 퍼지 논리 제어에서의 퍼지 자료를 조작하고 언어적 제어 규칙을 정의하는데 필요한 사항들을 정의한 데이터 베이스와 제어 전문가가 수행하는 일련의 제어 과정을 언어적 제어 규칙들로 나타낸 제어 규칙부로 구성된다. 그리고 비퍼지화부에서는 퍼지 출력값을 실제 제어 입력에 맞게끔 변환시켜 실제 제어 입력으로 사용할 수 있는 확정된 출력값으로 변환시켜 준다. 일반적으로, 제어 성능은 언어변수의 퍼지 집합의 선택, 소속 함수의 모양, 퍼지 법칙 베이스, 추론 방법, 비퍼지화 방법 등 여러 요인에 의해 영향을 받는다.

퍼지 제어기의 여러 구성 요소들을 최적으로 결정하는 많은 연구가 수행되어 왔으나. 비퍼지화기의 경우는 연산이 간단하고 그 역할에 대한 설계자들의 선입관 때문에 큰 주목을 받지 못했다. 퍼지 제어기의 비퍼지화기에 관한 몇 가지 연구 사례를 알아보면 다음과 같다. R. Yager와 D. Filev[2]는 SLIDE(Semi-Linear Defuzzification)방법이라고 불리는 파라메터화한 비퍼지화 연산자들의 집합을 소개하였다. 이 방법은 비퍼지화 연산이 제어기의 퍼지 출력 집합들의 간단한 선형 변환에 기초로 하고 있는데 SLIDE내 두개의 파라메터가 갖는 최적값을 Kalman 필터의 개념에 기초한 선형 방정식들의 최소 자승법에 의해 결정하고 있다. 그러나 사용된 알고리즘의 비선형성때문에 구해진 이들 파라메터값들이 국부해일 가능성이 있다. A. Bastian[3]는 퍼지 규칙들간의 비선형성을 제어할 수 있는 변형된 COA 비퍼지화 방법을 제안하였다. 이 비퍼지화 방법은 소속 함수간의 중첩되는 영역에 대한 비퍼지화 가중치(Defuzzification Weights)를 다층 퍼셉트론 신경망 구조상에서 역전파 학습 알고리즘에 의해 구하였다. 하지만 이 방법은 출력 변수들의 모든 소속 함수들이 같은 폭을 가지고, 같은 간격으로 위치하고 있는 경우만 사용 가능하다. 최근에, A. Ruiz[4]등은 구현시 하드웨어 복잡도가 큰 나눗셈을 사용하지 않고 모멘트 균형점을 찾는 것으로 비퍼지화 연산을 대신할 수 있는 방법을 제안하였다. 이 방법은 절단된 소속 함수가 가지는 영역을 충분히 반영하므로 정확한 비퍼지화 연산을 할 수 있지만, 왼쪽과 오른쪽 모멘트를 연속적으로 계산해야 하므로 비퍼지화값을 얻는데 많은 시간이 걸리는 단점이 있다.

본 논문은 비퍼지화기를 하드웨어로 구현시 하드웨어 복잡도를 줄이기 위해 출력 변수내 퍼지항들의 단일 지지값(Singletons)과 이에 대한 소속 함수값들의 가중치에 의해 결정되는 기존의 COA 비퍼지화 방법의 단점을 지적하고, 이를 개선하기 위해 소속 함수값 뿐 아니라 소속 함수의 폭(Span)을 동시에 고려한 영역 가중치(Aera Weights)를 사용함으로서 퍼지 논리 제어기의 제어 성능을 크게 향상시킬 수 있다는 것을 보인다. 나아가, 출력 변수가 갖는 소속 함수의 실효 폭(Effective Span)은 퍼지 제어기가 사용되는 문제에 매우 의존적이어서 주어진 문제에 가장 최적인 유효폭을 결정하는 것이 요구되는 데, 본 논문에서는 국부해 문제를 해결하기 위해 유전자 알고리즘을 사용하여 최적의 실효 폭을 결정한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 COA 비퍼지화 방법이 갖는 문제점을 기술하고, 3장에서는 이를 해결하기 위한 새로운 COA 비퍼지화 방법을 기술한다. 4장에서는 최적의 비퍼지화 가중치를 유전자 알고리즘에 의해 결정하는 방법을 설명하고, 5장에서는 제안한 COA 비퍼지화 방법을 트럭 후진

제어 문제에 적용하여 평균 주행 거리면에서 기존의 COA 비퍼지화 방법과 비교한다. 마지막으로 결론이 뒤따른다.

Ⅱ. 기존의 COA 비퍼지화 방법

비퍼지화는 퍼지 출력값을 실제 제어 입력으로 사용할 수 있는 확정된 출력값으로 변환시키는 과정이다. 기존의 비퍼지화기가 가지는 비퍼지화 방법에는 최대값 방법(Max Criterion), 최대 평균법(Mean Of Maximum), 그리고, 무게 중심법(Center of Area)이 있는데, 일반적으로 무게 중심법, 최대 평균법, 최대값 방법 순으로 좋은 성능을 나타낸다[5]. 퍼지 제어기를 하드웨어로 구현하는 데 무게 중심법이 가장 널리 사용되는데 그 이유는 다른 두 방법보다 더 좋은 Steady-state 제어 성능을 보이기 때문이다.[6]

COA 비퍼지화 방법은 비퍼지화값을 Max-Min 추론 엔진에 의해 얻어진 절단된(Clipped) 출력 변수가 갖는 소속 함수 값의 무게 중심으로 고려하는 방법으로 아래 식과 같이 나타내어진다.

$$y_c = \frac{\int_y y \cdot \mu_Y(y)\,dy}{\int_y \mu_Y(y)\,dy} \tag{1}$$

여기서 y는 추론된 소속 함수의 지지값, $\mu_y(y_i)$는 주어진 지지값에서 추론된 소속 함수의 최대값을 나타낸다.

COA 비퍼지화기에서 추론된 소속 함수를 최대 소속값에 대응하는 단일값(Singleton)으로 바뀌어도 만족할 만한 제어결과를 보이고[7] 이 경우 비퍼지화값 y_c는 아래 식과 같이 나타내어진다.

$$y_c = \frac{\sum_{i=1}^{n} y_i \cdot \mu_y(y_i)}{\sum_{i=1}^{n} \mu_y(y_i)} \tag{2}$$

여기서 n은 이산 퍼지항의 갯수, y_i는 i번째 퍼지 항의 단일 지지값, $\mu_Y(y_i)$는 단일 지지값 y_i에서의 소속 함수값을 나타낸다.

식 (1)과 식 (2)는 각 퍼지 항의 소속 함수가 대칭이고, 같은 폭을 가지고, 서로 같은 간격으로 배치될 때에만 동일한 비퍼지화값을 나타낸다. 그런데 아래 그림 2에서와 같이 위의 조건이 만족하지 않는 경우에는 식 (1)과 식 (2) 사이에 큰 오차가 초래된다. 아래 그림 2에서 a)와 b)의 경우 서로 동일한 두 개의 단일 지지값을 갖지만, 소속 함수들의 폭을 보면 a)의 경우는 서로 같고, b)의 경우는 서로 다름을 알 수 있다. 기존의 COA 비퍼지화기는 비퍼지화값 연산시 절단된 소속 함수값만을 고려하기 때문에 a)와 b)는 서로 같은 비퍼지화값을 나타낸다($y_c^a = y_c^b$). 그러나 이 결과는 면적이 더 넓은 곳 쪽으로 무게 중심이 이동해야 하는 일반적 사실과는 거리가 멀다. 이러한 문제점을 극복하기 위해 다음 장과 같은 새로운 비퍼지화 방법을 제안한다.

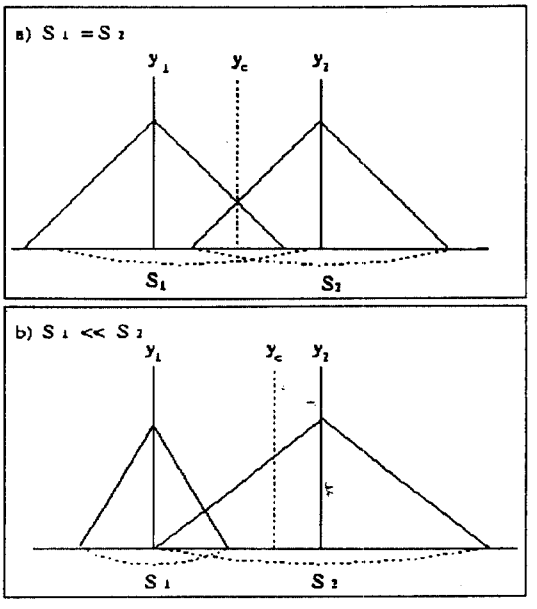


그림 2. 기존 COA 비퍼지화기의 오류의 예시
Fig. 2 A misuse of the conventional COA defuzzifier.

Ⅲ. 새로 제안한 COA 비퍼지화기

식 (1)의 분자와 분모는 y의 모든 출력 범위에서의 $y \cdot \mu_Y(y)$와 $\mu_Y(y)$의 영역을 나타내는데 비해 식 (2)는 특정한 소속 함수값이나 단일 지지값만을 고려하기 때문에 식 (1)과 식 (2)로부터 얻어지는 비퍼지화값

사이에는 뚜렷한 차이가 나타난다. 이를 해결하기 위해, 본 논문은 비퍼지화값을 다음과 같이 계산할 것을 제안한다.

$$y_c = \frac{\sum_{i=1}^{n} A_Y(y_i) \cdot y_i}{\sum_{i=1}^{n} A_Y(y_i)} \tag{3}$$

여기서 $A_Y(y)$는 i번째 추론된 소속 함수의 절단된 소속 함수값의 아래 영역, n은 퍼지항의 갯수, y_i는 i번째 퍼지항의 단일 지지값을 나타낸다.

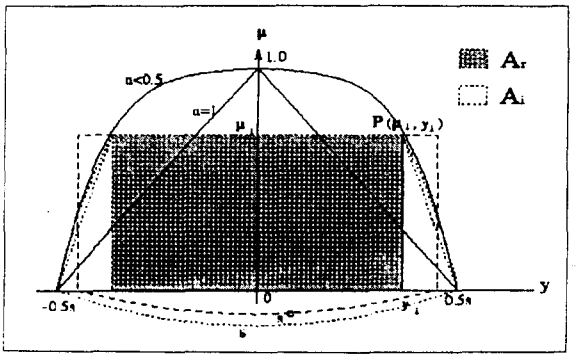


그림 3. 일반화된 소속 함수 $\mu_Y^n(y_i)$
Fig. 3 A generalized membership function $\mu_Y^n(y_i)$.

그림 3은 n<1 경우의 일반화된 소속 함수를 표현하는데 절단된 $\mu_y^n(y)$의 아래 영역 $A_Y(y_i)$는 다음과 같이 구할 수 있다.(식 (4)의 증명은 부록 1에 나타나 있다.)

$$A_Y(y_i) = \mu_Y(y_i) \cdot s_i - \frac{n}{n+1} \mu_Y^{\frac{n}{n+1}}(y_i) \cdot s_i$$

$$= \mu_Y(y_i) \cdot s_i \cdot (1 - \frac{n}{n+1} \mu_Y^{\frac{1}{n}}(y_i)) \tag{4}$$

여기서, $A_Y(y_i)$는 $n \to 0$이면 $\mu_Y(y_i) \cdot s_i$로 접근하고, $n \to \infty$이면 0으로 접근함을 알 수 있다. 여기서, 식 (4)의 $A_Y(y_i)$와 같은 면적을 갖는 가상적인 사각형을 생각한다. 그 사각형은 아래 식과 같이 높이 $\mu_Y(y_i)$, 폭 s_i^e로 된 유효 면적 $A_y^e(y_i)$를 갖는다고 볼 수 있다.

$$A_y^e(y_i) = \mu_y(y_i) \cdot s_i^e \tag{5}$$

여기서 s_i^e는 유효폭으로 다음과 같이 정의된다.

$$s_i^e = s_i \cdot (1 - \frac{n}{n+1} \mu_Y^{\frac{1}{n}}(y_i)) \tag{6}$$

이 경우에 비퍼지화값 y_c^e는 아래와 같이 구해진다.

$$y_c^e = \frac{\sum_{i=0}^{n} A_y^e(y_i) \cdot y_i}{\sum_{i=0}^{n} A_y^e(y_i)}$$

$$= \frac{\sum_{i=1}^{n} \mu_Y(y_i) \cdot s_i^e \cdot y_i}{\sum_{i=1}^{n} \mu_Y(y_i) \cdot s_i^e}$$

$$= \frac{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i \cdot (1 - \frac{n}{n+1} \cdot \mu_Y^{\frac{1}{n}}(y_i)) \cdot y_i}{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i \cdot (1 - \frac{n}{n+1} \cdot \mu_Y^{\frac{1}{n}}(y_i))} \tag{7}$$

시뮬레이션 분석으로부터 실효 폭 s_i^e는 퍼지 제어기를 어떤 문제에 응용하느냐에 따라 결정되며, 사용하는 소속 함수의 모양에도 영향을 받는다는 사실을 발견하였다. 그래서 본 본문에서는 특정 문제에 대해서 가장 최적인 유효폭을 얻을 수 있도록 상수 $\frac{n}{n+1}$를 비퍼지화 가중치 $w_i(0 \le w_i \le 1)$로 대신하였다. 이 경우, 앞서의 사각형은 아래 식과 같이 높이 $\mu_Y(y_i)$, 폭 s_i^o으로 된 최적 유효 면적 $A_y^o(y_i)$를 갖는다고 볼 수 있다.

$$s_i^0 A_y^o(y_i) = \mu_y(y_i) \cdot s_i^o \tag{8}$$

여기서는 최적의 유효폭으로 다음식과 같이 정의된다.

$$s_i^0 = s_i \cdot (1 - w_i \cdot \mu_Y^{\frac{1}{n}}(y_i)) \tag{9}$$

이 경우 비퍼지화값 y_c^o는 아래와 같이 구해진다.

$$y_c^o = \frac{\sum_{i=0}^{n} A_y^o(y_i) \cdot y_i}{\sum_{i=0}^{n} A_y^o(y_i)}$$

$$= \frac{\sum_{i=1}^{n} \mu_Y(y_i) \cdot s_i^o \cdot y_i}{\sum_{i=1}^{n} \mu_Y(y_i) \cdot s_i^o}$$

$$= \frac{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i \cdot (1-\omega_i \cdot \mu_Y^{\frac{1}{n}}(y_i)) \cdot y_i}{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i \cdot (1-\omega_i \cdot \mu_Y^{\frac{1}{n}}(y_i))} \quad (10)$$

소속 함수값 $\mu_Y(y_i)$가 $U(0, 1)$사이에 일정하게 분포한다고 가정하면 $\mu_Y^{\frac{1}{n}}(y_i)$의 기대값은 $\frac{n}{n+1}$ 이 되고, $n \leq 2$ 일 때 $\frac{n}{n+1} \cdot E[\mu_Y^{\frac{1}{n}}(y_i)] = (\frac{n}{n+1})^2 \ll 1$이므로 식 (4)에 있는 두 번째 항 $\frac{n}{n+1} \cdot \mu_Y^{\frac{1}{n}}(y_i)$은 무시할 수 있다. 이 경우 앞서의 사각형 면적은 근사적으로 $A_Y^a(y_i) = \mu_Y(y_i) \cdot s_i^a = \mu_Y(y_i) \cdot s_i$이 된다. 따라서 근사적 비퍼지화값 y_c^a는 다음과 같이 구해진다.

$$y_c^a = \frac{\sum_{i=0}^{n} A_Y^a(y_i) \cdot y_i}{\sum_{i=0}^{n} A_Y^a(y_i)}$$

$$= \frac{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i^a \cdot y_i}{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i^a}$$

$$= \frac{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i \cdot y_i}{\sum_{i=0}^{n} \mu_Y(y_i) \cdot s_i} \quad (11)$$

Ⅳ. 비퍼지화 가중치 w의 결정

비퍼지화 가중치 w는 그 분포 함수가 미리 알려져 있지 않으므로 선형 대수식을 풀어서 결정하기는 어렵다. 본 논문에서는 비퍼지화 가중치 w를 유전자 알고리즘에 의해 결정할 것을 제안한다. 유전자 알고리즘(Genetic Algorithm, GA)은 자연 선택과 생물학적 유전 현상에 기반한 반복적이고 적응적인 탐색 및 최적화 방법이다[8].

유전자 알고리즘은 다음과 같이 동작한다. P를 N개의 해개체(Chromosome)를 갖는 해집단, $P(0)$와 $P(t)$는 각각 무작위로 발생된 초기 해집단, 시간 t에서의 해집단이라고 하자. 그러면 $P(t)$에 일련의 복제(Reproduction), 교차(Crossover) 및 변이(Mutation) 연산을 가함으로서 새로운 해집단 $P(t+1)$를 얻게 된다. $P(t+1)$내 각 해개체는 시간 t에서 각 해개체가 얼마나 주어진 환경에 잘 적응하는 가를 나타내는 목적 함수(Fitness Function)의 값에 비례하여 재생된다. 교차 연산은 선택된 두 개의 해개체에 대해 임의로 정해진 교차점을 중심으로 유전 물질을 교환하는 방식으로 두 개의 해개체를 재결합시킨다. 변이 연산은 한 해개체내 임의로 선택된 개체(Gene) 값을 무작위로 변화시킨다. 이와 같은 과정을 여러 세대에 걸쳐 반복 시행하면 해집단의 평균 적합치는 점차로 증가하여 원하는 해로 접근하게 된다. 본 논문에서 사용한 유전자 알고리즘은 Goldberg등[9]이 사용한 이진 스트링을 갖는 유전자 알고리즘과 달리, 각 해개체내 개체값들이 실수를 갖는다. 사용된 유전자 알고리즘의 자세한 설명은 다음과 같다.

(1) 해개체의 표현 : 출력 변수가 N개의 퍼지항으로 구성되어 있는 경우, 비퍼지화 가중치 $w_i (i = 1, 2, \cdots, N)$는 N개의 연속적인 실수값으로 표현된다.

(2) 초기 해집단 발생 : 해개체내 비퍼지화 가중치 w의 초기값은 임의로 0과 1 범위내에서 N개의 실수를 발생함으로서 얻어진다. 초기 해집단은 위의 과정을 $|P|$반복함으로서 얻어진다. 여기서 $|P|$는 해집단내 해개체 수이다.

(3) 목적 함수 : 학습예 집합 $E = \{e^1, e^2, \cdots, e^T\}$내 한 학습예 $e^t (t = 1, 2, \cdots, T)$는 m개의 입력 값과 n개의 출력 값의 순서쌍 집합으로 구성된다. 두개의 입력 변수가 각각 x와 ϕ이고 하나의 출력 변수가 θ일 때 학습예 e^t는 $(e_x^t, e_\phi^t; e_\theta^t)$가 된다. 학습예 집합 E내 모든 학습예가 각 해개체에 대응하는 비퍼지화 가중치 w를 갖는 퍼지 제어기에 가해진다.

각 학습예 e^t에 대해서, 오차 ε^t는 $\varepsilon^t = \sum_{i=1}^{S} (y^t(i) - y_c^t(i))^2$로 계산된다. 여기서 S는 목적지까지 걸리는 스텝 수, $y^t(i)$는 학습예 집합 E로부터 미리 알려진 t번

째 학습예의 i번째 실제 출력값, 그리고 $y_c^t(i)$는 t번째 학습예에 대해 최적 COA 비퍼지화값을 나타내는 식 (9)에 의해 얻어진 i번째 확정된 값을 나타낸다. 전체 오차 E는 각 학습예에 의한 오차들의 합으로 $E=[\sum_{t=1}^{T}(\varepsilon^t)^2]^{\frac{1}{2}}$로 나타낸다. 이 경우, 각 해개체가 갖는 목적 함수 F는 $F=\frac{1}{1+k\cdot E}$로 나타낼 수 있으며, 여기서 k는 적당한 상수이다.

(4)유전자 연산:비퍼지화 가중치 w를 결정하는데 사용된 유전자 연산은 다음과 같다.

1)복제:새로운 해개체를 복제하기 위해 여러 가지 선택 방법을 혼합 사용한다. 첫 번째, 엘리트 선택(Elitist Selection)에 의해 가장 목적함수가 큰 해개체는 항상 다음 세대에 전달된다. 두 번째, 해집단내 새로운 해개체를 복제하기 위해 k-토너먼트 방법[10]을 약간 변형하여 사용한다. 이 방법은 목적 함수가 큰 상위 집단에서 임의로 k개의 해개체를 선택하여 그 중에서 가장 큰 목적 함수값을 갖는 해개체를 선택한다. 위의 절차를 두 번 연속하여 얻은 염색체 P_1과 P_2를 나중에 설명할 교차 및 변이 연산을 적용하여 새로운 염색체 C를 만든다. 새로운 해개체 C는 목적 함수값이 작은 하위 집단에서 임의로 k개를 선택하여 그 중에서 가장 작은 목적 함수값을 갖는 해개체 P'와 교체된다. 위와 같은 과정을 $pselect\times|P|$번(여기서 $|P|$는 해집단 크기임) 반복하여 새로운 해개체를 생성한다. 세 번째, 미성숙 수렴(Premature Convergence)을 피하기 위해 해집단내 일정 부분의 해개체들을 초기 해집단 발생 과정과 마찬가지로 임의의 값을 갖는 새로운 해개체를 발생시켜 채워 넣는다. 그림 4는 본 논문에서 사용되는 복제 과정을 나타낸 것이다.

2)교차:교차 연산은 아래 식과 같이 두 해개체 P_1과 P_2가 가지는 두 가중치 w_i^1와 w_i^2의 평균에 의해 얻어지는 새로운 비퍼지화 가중치 $w_i^c(i=1, 2, \cdots, N)$를 만든다.

$$w_i^c=\frac{w_i^1+w_i^2}{2} \tag{12}$$

3)변이:변이 연산은 다음과 같은 새로운 비퍼지화 가중치 w_i^m를 만든다.

$$w_i^m=\min(0,\ 1-w_i) \tag{13}$$

위의 식은 w_i가 $\frac{1}{2}$보다 큰 경우에 w_i^m는 w_i보다 $|w_i-\frac{1}{2}|$만큼 더 작다는 것을 의미하며, MIN 연산은 w_i^m가 음수가 되지 않도록 하기 위해서이다.

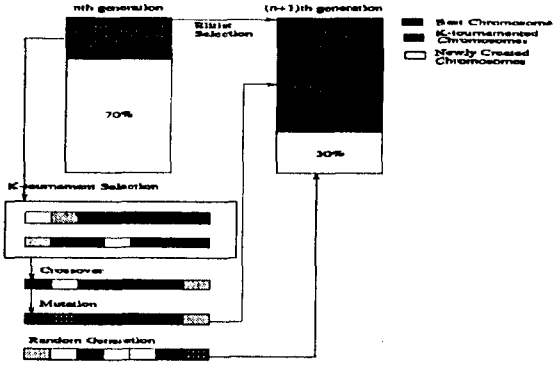


그림 4. 비퍼지화 가중치 연산에 사용된 복제 연산
Fig. 4 Reproduction used in determining defuzzification weights.

V. 시뮬레이션 결과 및 분석

3장에서 설명했던 COA 비퍼지화기들을 트럭 후진 제어 문제에 적용시켜 평균 주행 거리면에서 기존의 COA 비퍼지화기와 제어 성능을 비교해 본다. 트럭 주차 문제의 목표는 가능한 빨리, 그리고 정확하게 트럭을 주차시키는 것이며, 이 문제는 기존 제어 기술로는 풀기 힘든 전형적인 비선형 제어 문제이다. 그림 5는 트럭 주차 제어 문제에서 사용된 트럭과 주차대의 위치를 보여준다. 트럭의 위치는 (x, y, ϕ)에 의해 결정된다. 단, 여기에서 ϕ는 트럭 진행 방향과 x 축간의 각도이며, 트럭의 후진 주행 제어는 ϕ와 핸들의 축 간의 각도인 θ에 의해 결정된다. 트럭이 움직이는 운동 방정식은 아래와 같이 나타내어진다[11].

$$\begin{aligned}x(t+1)&=x(t)+\cos[\phi(t)+\theta(t)]+\sin[\theta(t)\cdot\sin[\phi(t)]\\ y(t+1)&=y(t)+\sin[\phi(t)+\theta(t)]-\sin[\theta(t)\cdot\sin[\phi(t)]\\ \phi(t)&=\phi(t)-\sin^{-1}[2\sin(\frac{\theta(t)}{b})]\end{aligned} \tag{14}$$

여기서 b는 트럭의 길이이며, 본 논문에서는 b=4로 하였다.

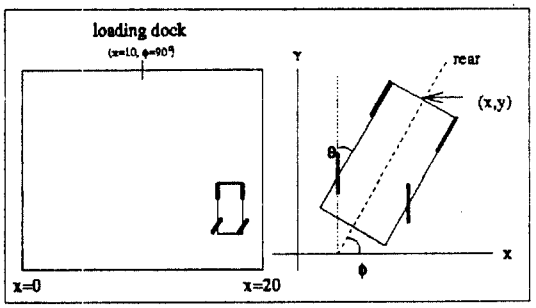


그림 5. 모형 트럭과 주차대 위치
Fig. 5 A model truck and loading dock.

만약 트럭과 주차대까지의 거리가 충분하다면 트럭이 $x=10$, $\phi=90°$ 가까이 오면 트럭을 곧장 후진하기만 하면 되기 때문에 변수 y를 퍼지 입력 변수 (x, y, ϕ)에서 뺄 수 있다. 그러므로 트럭 주차 제어 문제는 주어진 공간내 $\{0\le x\le 20,\ -90°\le\phi\le 270°\}$ 임의의 초기 위치 (x_0, ϕ_0)에서 가능하면 신속·정확하게 주차대 ($x=10$, $\phi=90°$) 쪽으로 후진하도록 바퀴 각도 θ $(-40\le\theta\le 40)$를 제어하는 것이 요구된다. 그림 6과 7은 모양 상수 $n=1$ 일 때, 퍼지 제어기의 입·출력 변수의 소속 함수와 퍼지 제어를 위한 퍼지 규칙 베이스를 나타낸 것이다. 이 두 데이터는 Wang과 Mendel의 논문에서 참조하였다.[12]

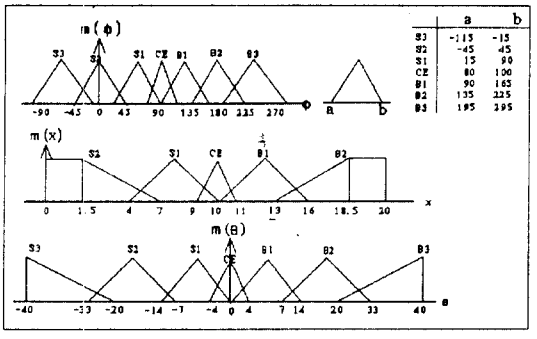


그림 6. 트럭 후진 주차 문제에 사용된 소속 함수
Fig. 6 Membership functions for the truck backer-upper control problem.

Φ \ X	S2	S1	CE	B1	B2
S3	S2	S3			
S2	S2	S3	S3	S3	
S1	B1	S1	S2	S3	S2
CE	B2	B2	CE	S2	S2
B1	B2	B3	B2	B1	S1
B2		B3	B3	B3	B2
B3				B3	B2

그림 7. 트럭 후진 주차 문제에 사용된 퍼지 규칙 베이스
Fig. 7 Fuzzy rule base for the truck backer-upper control problem.

3가지 서로 다른 COA 비퍼지화값 y_c^e, y_c^o 및 y_c^a은 각각 식 (7), (10) 및 (11)로부터 얻어지며, 기존의 COA 비퍼지화값 y_c은 식 (2)로부터 얻어진다. 이 비퍼지화기들을 사용하는 퍼지 제어기들의 평균 주행 거리를 비교하여 얼마나 제어 성능을 향상시키는지를 알아본다. 제어 성능은 1,000개의 임의로 선택된 초기 위치값 (x_i, ϕ_i)로부터 출발하는 트럭의 평균 주행 거리에 의해 나타내어진다. 식 (10)의 y_c^o를 비퍼지화값으로 사용하는 경우, 비퍼지화 가중치 w는 4장에서 설명했던 유전자 알고리즘에 의해 얻어진다. 유전자 알고리즘을 위해 사용되는 학습 데이터는 14개의 초기 위치로부터 주차대까지 부드러운 주행 경로를 나타내는 학습예 집합을 이용한다.(실제 데이터는 [12]에 있는 표 1부터 표 14까지 참조.) 비퍼지화 가중치 w를 결정하기 위해 사용된 유전자 알고리즘의 실행 패라메터는 해집단 크기 $|P|=200$, 해개체 크기 $|w|=7$, 토너먼트 $k=3$, $pselect=0.7$, 교차 확률 $P_c=1.0$, 변이 확률 $P_m=0.001$, 세대수 $N=2000$으로 주어진다. 그림 8은 유전자 알고리즘이 수행될 때 얻어지는 목적 함수의 진화 곡선을 보여주는 것으로 세로축의 값은 해집단내 모든 해개체들에 의한 목적함수의 평균값을 나타낸다. 이 그림으로부터, 세대수가 증가하면서 평균 목적 함수값이 천천히 증가하는 것을 볼 수 있다.

표 1은 유전자 알고리즘을 사용하여 결정된 최적의 비퍼지화 가중치 w를 나타낸다. 전체 비퍼지화 가중치의 평균으로 정의되는 w_{global}은 모양 상수 n이 커질수록 증가한다. 이는 소속 함수의 실효 면적이 모양 상수 n이 커질수록 작아지는 것을 상쇄시키기 위한

것으로 해석된다.

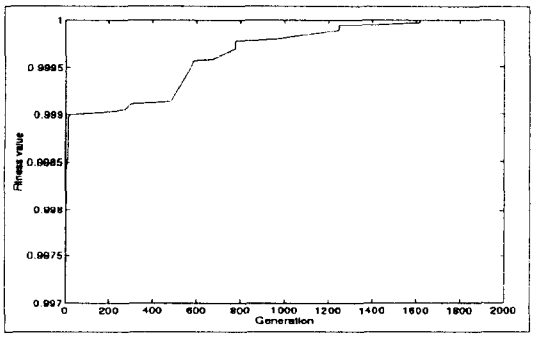


그림 8. 목적 함수값의 진화 곡선.
Fig. 8 A evolution curve of the fitness function value.

표 1. 유전자 알고리즘으로부터 얻은 비퍼지화 가중치
Table 1. Defuzzification weights obtained from the GA.

Weighls	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_{global}
n = 0.5	0.705	0.652	0.618	0.618	0.633	0.615	0.554	0.628
n = 1.0	0.732	0.693	0.652	0.652	0.669	0.651	0.605	0.605
n = 2.0	0.761	0.712	0.682	0.676	0.679	0.697	0.631	0.691

그림 9는 임의의 1,000개 시작점에 대해 모양 상수가 1.0인 여러 가지 비퍼지화기가 나타내는 평균 주행 거리를 보인 것이다. 평균 주행 거리가 최적 COA 비퍼지화기(y_c^o), 유효 COA 비퍼지화기(y_c^e), 근사 COA 비퍼지화기(y_c^a), 및 기존 COA 비퍼지화기(y_c) 순으로 각각 20.07 단계, 20.89 단계, 21.41 단계, 및 25.48 단계 순으로 짧아짐을 알 수 있다. 이것은 최적 COA 비퍼지화기를 갖는 퍼지 제어기가 기존의 COA 비퍼지화기를 갖는 퍼지 제어기보다 주행 거리면에서 평균적으로 20% 이상 짧아짐을 나타낸다. 이들 그림들로부터, 주행 거리가 짧아지는 것은 주로 진행 방향이 급격히 변화하는 근처에서 주행 각도 θ의 정확한 제어에 의해 얻어짐을 알 수 있다. 즉 그림 6의 θ에 대한 소속 함수를 보면 θ값이 양쪽으로 커질수록 소속 함수의 폭이 커지는 데 기존의 COA 비퍼지화기는 비퍼지화값 계산시 이를 효과적으로 반영하지 못하므로 진행 방향이 급격히 변화하는 근처에서(즉 θ값이 큰 영역) 정확한 제어를 하지 못하여 전체적으로 주행거리가 증가시키는 결과를 초래한다.

나아가, 근사 COA 비퍼지화기의 경우 비퍼지화값 연산시 출력 변수의 퍼지항의 단일 지지값에 절단된 소속 함수값과 소속 함수의 원래 폭을 곱한 값이 가중치로 작용하므로 제안한 다른 두 가지 COA 비퍼지화기에 비해 하드웨어 복잡도가 훨씬 줄어든다. 그럼에도 불구하고, 근사 COA 비퍼지화기를 사용한 퍼지 제어기의 제어 성능은 다른 두 가지 COA 비퍼지화기를 사용한 퍼지 제어기들과 비교해 보면 그리 많이 떨어지지 않으며, 기존의 COA 비퍼지화기를 갖는 퍼지 제어기보다 훨씬 개선된 제어 성능을 나타낸다. 따라서, 비용 대비 성능 관점에서 보면, 근사 COA 비퍼지화기를 갖는 퍼지 제어기가 가장 유리하다고 볼 수 있다.

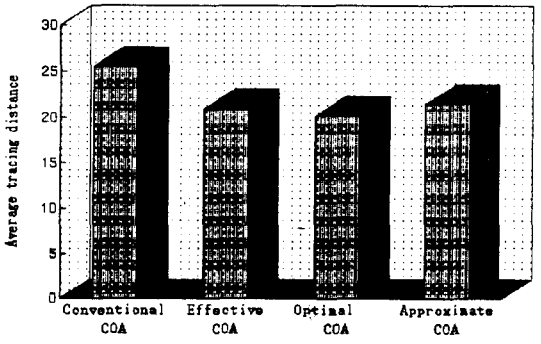


그림 9. 네 가지 다른 COA 비퍼지화기를 갖는 퍼지 제어기의 제어 성능
Fig. 9 Comparison of average tracing distances of four different FLCs.

그림 10은 임의의 두 위치 {(19.1, 0.0, 206°), (2.3, 0.0, −34.6°)}에서 출발한 트럭에 대해, 서로 다른 COA 비퍼지화기를 사용하는 네 개의 퍼지 제어기들에 의해 얻어진 주행 경로를 보여 준다. 이 경우, 주행 거리가 최적 COA 비퍼지화기(y_c^o), 유효 COA 비퍼지화기(y_c^e), 근사 COA 비퍼지화기(y_c^a), 및 기존 COA 비퍼지화기의 순으로 (19.1, 0.0, 206°)의 경우 각각 18, 19, 20 및 26 단계가 걸렸으며, (2.3, 0.0, −34.6°)인 경우 각각 19, 20, 21 및 26 단계가 걸렸다.

표 2는 모양 상수 n이 4가지 COA 비퍼지화기를 사용하는 퍼지 제어기의 제어 성능에 어떠한 영향을 미치는가를 알아본 것이다. 이 표로부터 일반적으로 제

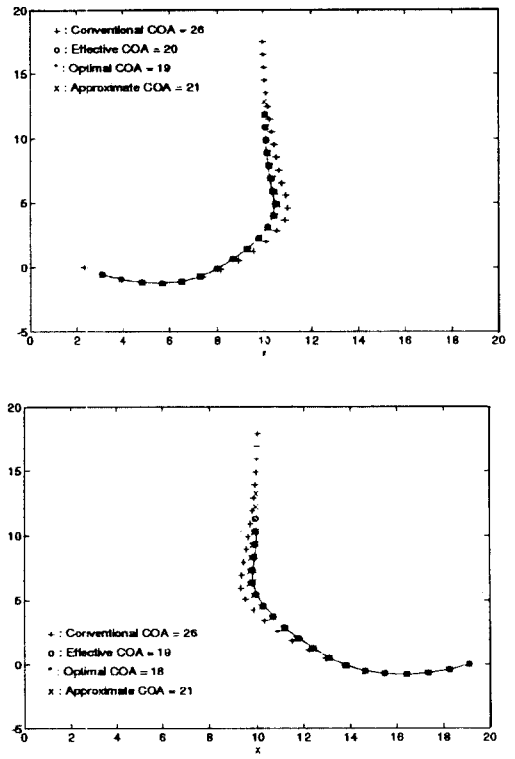


그림 10. 네 가지 다른 COA 비퍼지화기를 갖는 퍼지 제어기의 주행 경로

Fig. 10 Comparison of tracing trajectories of four different FLCs.

표 2. 서로 다른 모양 상수 n을 갖는 퍼지제어기의 제어 성능

Table 2. Comparison of control performance of different shape constants n

Shape Constant	Conventional COA	Effective COA	Optimal COA	Approximate COA
n = 0.5	25.48 steps	21.11 steps	20.54 steps	21.41 steps
n = 1.0	25.48 steps	20.89 steps	20.07 steps	21.41 steps
n = 2.0	25.48 steps	20.69 steps	19.85 steps	21.41 steps

어 성능은 모양 상수 n이 커질수록 좋아진다는 사실을 발견하였다. 이것은 모양 상수 n이 커질수록 중첩되는 영역의 크기가 작아져 가중치가 제어 성능에 미치는 영향이 상대적으로 증대되기 때문이다. 최적 COA 비퍼지화기(y_c^o)를 사용한 퍼지 제어기와 실효 COA 비퍼지화기(y_c^e)를 사용한 퍼지 제어기의 평균적인 제어 성능은 큰 차이가 없다. 하지만 임의의 초기 위치에 대해 실효 COA 비퍼지화기(y_c^e)를 사용한 퍼지 제어기는 최적 COA 비퍼지화기(y_c^o)를 사용한 퍼지 제어기보다 목적지에 도달하지 못하는 경우가 더 많이 발생된다.

Ⅵ. 결 론

본 논문은 보다 정확한 제어 성능을 보이는 퍼지 논리 제어기를 위한 새로운 COA 비퍼지화기를 제안하였다. 기존의 COA 비퍼지화기를 하드웨어로 구현시 하드웨어 복잡도를 줄이기 위해 흔히 절단된 소속함수값과 퍼지항의 단일 지지값을 고려한다. 따라서 기존의 COA 비퍼지화 방법은 단일 지지값들이 서로 같을 경우에는 소속 함수들의 폭이 다름에도 항상 같은 비퍼지화값을 보이는 잘못된 결과를 나타낸다. 이런 오류를 극복하기 위해, 비퍼지화값 연산시 소속함수값과 소속 함수의 폭의 곱으로 정의되는 실효 면적 개념을 도입하였다. 실효 면적은 적용하는 제어 대상과 사용하는 소속 함수의 형태에 매우 의존적이어서 주어진 문제에 최적인 실효 면적을 고려하기 위해 비퍼지화 가중치 w를 도입하였고, 주어진 문제에 가장 최적인 비퍼지화 가중치 w를 학습예를 통한 유전자 알고리즘을 사용하여 결정하였다.

본 논문의 가장 중요한 결과는 다음과 같다. (1) 비퍼지화값을 연산할 때 소속 함수의 최적폭을 고려함으로서 20% 이상의 제어 성능을 향상시켰고, (2) 최적의 비퍼지화 가중치 w를 유전자 알고리즘을 사용함으로서 쉽게 결정할 수 있었고, (3) 소속 함수의 최적폭을 구하는 대신 원래 소속 함수가 갖는 폭을 대신하여도 얻어진 제어 성능이 최적폭을 사용할 때보다 그리 많이 떨어지지 않는다는 것을 알았다는 점이다. 그러나 제안한 여러 COA 비퍼지화 방법은 실효 면적을 계산하는데 더 많은 하드웨어를 필요로 한다. 이런 부가적인 하드웨어의 사용을 배제하기 위해 곱셈기 및 나눗셈기를 사용하지 않고 비퍼지화를 수행하는 연구가 진행되고 있다.

참 고 문 헌

1. S. K. Halgamuge and M. Glesner, "Fuzzy Neural Fusion Techniques for Industrial Application,"

ACM Symposium on Applied Computing (SAC'94), Phoenix, USA, March 1994.

2. R. R. Yager and D. P. Filev, "SLIDE: A Simple Adaptive Defuzzification Method," *IEEE Transaction on Fuzzy Systems*, vol. 1, no. 1, pp. 69-78, Feb. 1993.
3. A. Bastian, "Handling the nonlinearity of a fuzzy logic controller at the transition between rules," *Fuzzy Sets and Systems*, no. 71, pp. 369-387, 1995.
4. A. Ruitz, J. Gutiérrez, and J. Fernández, "A Fuzzy Controller with an Optimized Defuzzification Algirithm," *IEEE Micro*, pp. 1-10, Dec. 1995.
5. C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404-435, Feb. 1990.
6. T. A. Runkler and M. Glesner, "A set of Axioms for Defuzzification Strategies Towards a Theory of Rational Defuzzification Operators," *Second IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1161-1166, 1993.
7. T. Miki, H. Matsumoto, K. Ohto, and T. Yamakawa, "Silicon Implementation for a Novel High Speed Inference Engine: Mega-FLIPS Analog Fuzzy Processor," *Journal of Intelligent and Fuzzy Systems*, vol. 1, no. 1, pp. 27-42, 1993.
8. J. H. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michigan Press, 1975.
9. David E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," *Addison-Wesley Press*, 1989.
10. David E. Goldberg and Kalyanmoy Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," Ed. Gregory J. E Rawlins, *Fundamental of Genetic Algorithms*, Morgan Kaufmann Publisher, pp. 69-93, 1991.
11. Li-Xin Wang And Jerry M. Mendel, "Generating Fuzzy Rules from Numerical Data with Applications," *USC-SIPI Report*, no. 169, 1991.
12. Li-Xin Wang and Jerry M. Mendel, "Generating Fuzzy Rules by Learning from Examples," *IEEE Transactions on System, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414-1427, Nov. 1992.

부록 1: 식 (4)의 유도

그림 3에서, 절단된 소속 함수 아래의 영역 $A_Y(y_i)$은 사각형 영역 A_r과 적분 영역 A_i의 합이다. 그리고 소속 함수가 갖는 포물선 방정식 $\mu_Y(y_i)$는 $\mu_Y(y_i)=m\cdot(y_i-\frac{s_i}{2})^n$로 나타내어진다. 상수 m은 조건 $\mu_Y(0)=1$을 이용하면 $m\cdot(0-\frac{s_i}{2})^n=1$로부터 $m=(\frac{s_i}{2})^{-n}$이다.

따라서, 포물선 방정식 $\mu_Y(y_i)$는 $\mu_Y(y_i)=(-\frac{s_i}{2})^{-n}\cdot(y_i-\frac{s_i}{2})^n=(1-\frac{2y_i}{2})^n$로 쓸 수 있다. 교차점 $P(y_i, u_i)$에서 $\mu_Y(y_i)=\mu_i$이므로, $(1-\frac{2y_i}{s_i})^n=\mu_i$로부터 y_i가 $\frac{s_i}{2}\cdot(1-\mu_Y(y_i)^{\frac{1}{n}})$임을 알 수 있다.

사각형 영역 A_r는

$$\begin{aligned}A_r&=2\cdot\mu_i\cdot y_i\\&=2\cdot\mu_Y(y_i)\cdot y_i\\&=s_i\cdot(1-\mu_Y(y_i)^{\frac{1}{n}})\cdot\mu_Y(y_i)\end{aligned}$$

이 되고, 적분 영역 A_i는

$$\begin{aligned}A_i&=2\cdot\int_{y_i}^{\frac{s_i}{2}}(1-\frac{2y}{s_i})^n\,dy\\&=2[-\frac{s_i}{2}\cdot\frac{1}{n+1}\cdot(1-\frac{2y}{s_i})^{n+1}]_{y_i}^{\frac{s_i}{2}}\\&=\frac{s_i}{n+1}\,\mu_Y^{\frac{n+1}{n}}(y_i)\end{aligned}$$

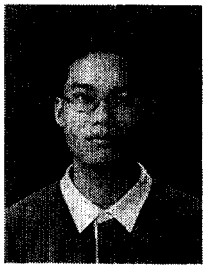
이 된다. 그러므로 전체 영역 $A_Y(y_i)$는

$$\begin{aligned}A_Y(y_i)&=A_r+A_i\\&=s_i\cdot(1-\mu_Y^{\frac{1}{n}}(y_i))\cdot\mu_Y(y_i)+\frac{s_i}{n+1}\,\mu_Y^{\frac{n+1}{n}}(y_i)\\&=s_i\cdot\mu_Y(y_i)-s_i\cdot\frac{n}{n+1}\cdot\mu_Y^{\frac{n+1}{n}}(y_i)\\&=s_i\cdot\mu_Y(y_i)-(1-\frac{n}{n+1}\cdot\mu_Y(y_i))\end{aligned}$$

이 된다.

$n>1$인 경우에도 같은 방법으로 증명이 된다. 이로써 식 (4)의 증명을 끝낸다.

조 인 현(In-Hyun Cho) 준회원
1996년 2월 : 동아대학교 전산공학과 졸업
1996년~현재 : 동아대학교 대학원 컴퓨터공학과 석사과정
※주관심분야 : 소프트웨어 공학, 소프트 컴퓨팅, VLSI/ASIC 설계 등.



이 동 석(Dong-Seog Lee)준회원
1995년 2월 : 동아대학교 전산공학과 졸업
1995년~현재 : 동아대학교 대학원 컴퓨터공학과 석사과정
※주관심분야 : H/W & S/W Codesign, 소프트 컴퓨팅 등.

김 종 훈(Jong-Hoon Kim) 정회원
1974년 2월 : 동아대학교 전자공학과 졸업
1977년 2월 : 동아대학교 대학원 전자공학과 졸업(공학석사)
1986년 2월 : 경북대학교 대학원 전자공학과 졸업(공학박사)
1986년~현재 : 동아대학교 컴퓨터공학과 교수.
※주관심분야 : Inteligent System Design, Testing, Real-time System S/W, H/W & S/W Codesign 등.

김 대 진(Daijin Kim) 정회원
1981년 2월 : 연세대학교 전자공학과 졸업
1984년 2월 : KAIST 전기 및 전자공학과 졸업(공학석사)
1991년 8월 : Syracuse 대학 ECE 졸업(공학박사)
1984년 3월~1987년 2월 : KBS 기술연구소 HDTV팀
1992년~현재 : 동아대학교 컴퓨터공학과 조교수
※주관심분야 : 화상 처리, 소프트 컴퓨팅, VLSI/ASIC 설계 등.