

Schema Co-Evolutionary Algorithm을 이용한 2-Layer Fuzzy Controller의 최적 설계

Optimal Design of the 2-Layer Fuzzy Controller using the Schema Co-Evolutionary Algorithm

심귀보 · 변광섭

Kwee-bo Sim and Kwang-Sub Byun

중앙대학교 전자전기공학부

요 약

최근 들어, 다양하고 복잡한 기능을 갖는 로봇이 요구되고 있다. 그러나 이전의 알고리즘으로는 그러한 요구를 만족시킬 수 없다. 이러한 문제를 해결하기 위해, 본 논문에서는 다양한 입력과 출력을 다루는 경우에도 작은 개수의 퍼지 룰을 갖고, 효율적이고 강인하게 제어할 수 있는 2-Layer Fuzzy Controller를 소개한다. 그런데 퍼지 제어기에서의 중요한 문제점은 퍼지 룰 베이스를 어떻게 설계하는지에 달려 있다. 본 논문은 Schema Co-Evolutionary Algorithm을 이용하여 최적의 2-Layer Fuzzy Controller를 설계하는데, 이 Schema Co-Evolutionary Algorithm은 simple GA보다 더 빠르고 우수하게 최적해를 찾을 수 있다.

Abstract

Nowadays, the robot with various and complex functions is required. Previous algorithms, however, cannot satisfy the requirement. In order to solve these problems, we introduce the 2-Layer Fuzzy Controller, which has a small number of fuzzy rules corresponding to various inputs and outputs. Also, it controls robustly and effectively an object. The main problem in the fuzzy controller is how to design the fuzzy rule. This paper designs the optimal 2-layer fuzzy controller using the Schema Co-Evolutionary Algorithm. The schema co-evolutionary algorithm can find more rapidly and excellently than simple genetic algorithm does.

Key words : 2-layer fuzzy controller, schema co-evolutionary algorithm, robot following, avoidance of collision

1. 서 론

기계/전자 공학 기술을 기반으로 네트워크의 급속한 발전으로 인해 인간을 닮은 휴머노이드 로봇에 관한 연구가 활발히 진행 중이다. 특히 일부 기술-제어/계측 분야, 감지 분야, 2족 보행 기술 등-에서는 상당한 수준으로 발전하고 있다. 여기에서 앞으로 연구되어야 할 부분은 인공지능 분야이다. 즉 로봇이 다양한 정보를 가지고 주변 환경을 스스로 판단하며 행동할 수 있는 알고리즘에 관한 연구가 필요하다. 로봇에게 지능을 부여하는 알고리즘이 많이 제안되어 왔으며, 퍼지 시스템, 인공 신경망, 진화 연산 등이 그 대표적인 예이다.

로봇을 제어하는 데 있어서 가장 많이 사용되는 것은 퍼지 제어기이다. 그 이유는, 퍼지 제어 시스템이 제어조절능력의 자유도가 높고 파라미터 변화나 잡음에 대한 민감도가 낮아 견실하며 실제적으로 활용하기 쉽고 범용성이 높다는 장

점을 갖고 있기 때문이다 [1]. 그림 1은 이러한 퍼지 제어기의 구조를 보여준다. 퍼지 엔코더는 퍼지 제어의 프로그램을 위한 인터페이스로 측정된 입력정보의 양자화 작업이나 퍼지 추론을 위한 수치처리문제를 다루는 부분이다. 퍼지 제어기의 규칙기반은 전건부(입력변수 조건)와 후건부(출력변수 결과)로 나타내며, 퍼지 제어에서는 이 부분이 가장 중요하다. 퍼지 추론 엔진에서는 퍼지 규칙을 이용하여 입력에 대해 적절한 출력을 만들어낸다. 마지막으로 퍼지 디코더는 퍼지 추론결과를 프로세스의 제어에 맞도록 수치를 변환하는 작업과 퍼지 변수를 실제 값으로 바꾸는 디퍼지화 과정을 포함한다.

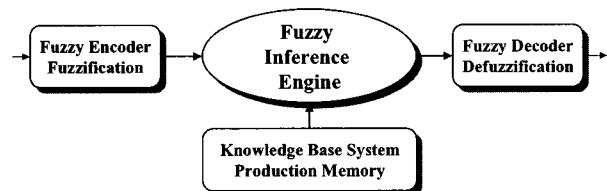


그림 1. 퍼지 제어 시스템의 구조
Fig. 1. Configuration of fuzzy control system

접수일자 : 2004년 1월 14일

완료일자 : 2004년 3월 31일

본 연구는 과학기술부의 뇌신경정보학연구사업의 '뇌정보 처리 메커니즘에 기반한 인간행동시스템연구'의 연구비 지원으로 수행되었습니다. 연구비 지원에 감사드립니다.

그런데, 퍼지 제어기는 입력의 개수가 늘어남에 따라 퍼지 룰의 개수가 지수적으로 증가한다는 단점을 갖고 있다. 퍼지 룰의 개수의 증가는 퍼지 제어기의 크기를 지수적으로 크게 할 뿐만 아니라, 퍼지 제어기를 설계하는 데에도 큰 어려움이 따르게 된다. 요즘과 같이 로봇의 기능이 다양하고 복잡해지는 경우에 있어서 이러한 퍼지 룰의 증가는 큰 제약이 된다. 이러한 문제를 해결하기 위하여 2-Layer Fuzzy Controller (2LFC)가 제안되었다 [2].

퍼지 이론을 처음으로 제안한 Zadeh에 의하면, 퍼지 룰은 surface 구조이거나 deep 구조를 가질 수 있다 [3]. Surface 구조는 퍼지 제어기를 적용할 대상에 따라 달라지고 언어적인 변수로 표현된다. 이 구조를 통해 퍼지 제어기가 어떻게 동작할 지를 대략적으로 알 수 있다. 하지만 surface 구조는 퍼지 변수의 파라미터들이 결정되어 있지 않기 때문에 각각의 퍼지 변수를 결정해 주어야 한다. 퍼지 변수를 결정하는 방법에는 여러 가지가 있으며 그림 2는 Zadeh가 언급한 방법들을 보여준다. 이렇게 해서 퍼지 변수가 결정되어 완전한 퍼지 룰이 결정되면 비로소 전체 퍼지 제어기가 결정된다. 즉 퍼지 제어기를 설계하는데 있어서 가장 중요한 것은 퍼지 룰을 설계하는 것이며, 퍼지 룰을 설계한다는 것은 적용할 대상에 따른 surface 구조를 deep 구조로 바꿔주는 것이다.

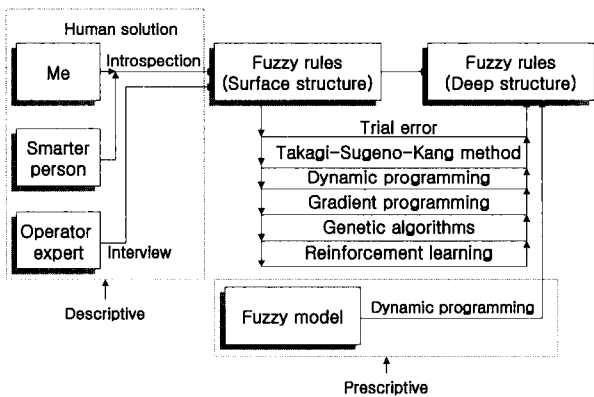


그림 2. Zadeh가 제안한 퍼지 룰의 deep 구조 유도 방법
Fig. 2. Summary of alternative methods to deduce the deep structures of a set of rules, proposed by Zadeh

퍼지 룰을 설계하는 여러 기법들 중에서 최근에 많은 관심을 받고 있는 것은 진화 연산이다. 진화 연산의 대표적인 유전자 알고리즘(genetic algorithm : GA)은 자연세계의 진화과정을 컴퓨터상에서 시뮬레이션 함으로써 복잡한 실세계의 문제를 해결하고자 하는 계산 모델이다. 또한 GA는 구조가 간단하고 방법이 일반적이어서 응용범위가 매우 넓으며, 특히 적응적 탐색과 학습 및 최적화를 통한 공학적인 문제의 해결에 많이 이용되고 있다. 그런데, GA가 모든 최적화 문제에 좋은 결과를 가져오는 것은 아니다. 즉 최적해를 찾지 못하고 국소해에 머무는 경우가 많다. 그 이유는 최적해를 포함하는 스키마의 적합도가, 경쟁하는, 최적해를 포함하지 않는 스키마의 적합도보다 낮아서 결국 최적해를 포함하지 않는 스키마 쪽으로 수렴하기 때문이다 [4]. 즉 GA는 하나의 개체군으로 이루어져 있어서, 미리 정의되어지는 적합도 함수에 의하여 성능의 제약을 받게 된다. 이와 달리 자연계는 두 개체군간의 상호작용과 공진화를 통하여 동적인 적합도 환경을 제공받는다. 이러한 관점에서 공진화에 대한 관심이

높아지고 있다.

GA의 단점을 해결하는 공진화 알고리즘으로 스키마 공진화 알고리즘 (schema co-evolutionary algorithm : SCEA)이 제안되었다 [4][5]. 이 알고리즘은 스키마로 구성된 부가적인 개체군을 도입하여 기존의 해의 후보들로 구성된 주 개체군과의 공진화를 통해, 최적해를 포함하면서 평균 적합도가 낮은 스키마들도 기존의 GA에서보다 개체수의 기대치를 높여서 선택 될 가능성을 높였다.

본 논문에서는, 다양한 입력 정보를 가지고 다양한 행동을 출력하는 로봇을 제어하기 위한 알고리즘으로 2-Layer Fuzzy Controller (2LFC)를 사용하고, 이 퍼지 제어기를 설계하기 위하여 스키마 공진화 알고리즘 (SCEA)을 이용한다. 본문의 2장에서는 2LFC의 구조와 실제 적용에 대해 설명한다. 3장에서는 이러한 2LFC를 설계하기 위해 적용된 SCEA에 대해 설명하고 이것을 2LFC에 적용하는 기법을 소개한다. 4장에서는 로봇제어를 위한 2LFC의 설계에 적용된 SCEA와 SGA를 비교하였으며, SCEA를 이용한 시뮬레이션을 통해 최적의 2LFC를 설계하고, 실제 실험을 통해 2LFC의 효율성을 입증한다. 마지막으로 5장에서 본 논문의 결론에 대해 논한다.

2. 2-Layer Fuzzy Controller

2.1. 2-Layer Fuzzy Controller의 구조

일반적인 퍼지 제어기에서는 입력의 개수가 증가함에 따라 전체 퍼지 룰의 개수는 지수적으로 증가한다. 이에 따라, 조정해야 할 시스템 파라미터도 지수적으로 증가한다. 이러한 두 가지 관점은 차원문제(dimensionality problem)라고 불린다 [6]. 따라서 일반적인 퍼지 제어기로 다양한 입력 또는 다양한 출력을 갖는 로봇을 제어하는 데는 많은 어려움이 따른다. 이러한 문제를 해결하기 위하여 계층적 퍼지 시스템의 일종인 다단계 퍼지 신경망 모델이 Chung과 Duan에 의해 제안되었다. 그들은 계층적 퍼지 시스템을 세 가지 즉, 증가형, 집합형, 종속형으로 분류하였다 [7].

이와 비슷하게, 다양한 입력을 가지면서 다양한 행동을 출력하는 로봇을 제어하는데 효율적인 알고리즘으로 2-Layer Fuzzy Controller(2LFC)가 제안되었다 [2]. 그림 3은 2LFC의 구조를 나타낸다. 특히, 본 그림의 2LFC는 첫 번째 layer에 4 개의 sub-controllers를 갖고 두 번째 layer에서 2 개의 combined controller를 갖는다. 첫 번째 layer에서 다양한 입력들이 네 가지 형태로 분류되어 sub-controllers로 입력된다. Sub-fuzzy controllers는 적절한 퍼지 추론 시스템을 이용하여 독립적으로 퍼지 추론을 한다. 그 다음, 두 번째 layer는 combined controllers의 입력으로, sub-controllers의 출력을 사용한다. 이때, 각각의 combined controller 다른 형태의 최종 출력을 위해서 첫 번째 layer의 출력을 필요로 한다. 두 번째 layer의 combined controllers는 통합적인 퍼지 추론을 한다. 그러나 최종의 최적의 출력을 위해 각각의 combined controller는 서로 배타적이지 않다. 이러한 과정을 거쳐, 2LFC는 다양한 입력과 다양한 출력을 가지고 퍼지 추론을 수행한다.

그림 3에서 보듯이, 2LFC는 다단계 퍼지 추론 시스템의 집합형 구조와 비슷하다. 또한 신경망 모델과도 비슷해 보인다. 그러나 2LFC는 세 가지 다른 점이 있다. 첫째로, 2LFC는 다단계 퍼지 시스템의 특수한 경우로서, 퍼지 추론의 단계를 2단계까지만 한정한다. 다단계 퍼지 시스템의 경우에는

늘어나는 단계에 대한 해결책이 분명하지 않아서 전체 시스템의 안정성을 보장할 수 없다. 그러나 2LFC는 퍼지 추론의 단계를 2단계로만 한정함으로써 내부적인 시스템의 안정성을 확보할 수 있다. 둘째로, 2LFC는 최종 출력이 연결된 마지막 단계에서 둘 이상의 제어를 갖는다. 요즘에는 제어해야 할 대상이 매우 복잡하고 다양하기 때문에, 다양하고 많은 입력뿐만 아니라, 다양하고 많은 출력을 제어하는 것이 요구된다. 다단계 퍼지 시스템은 많은 입력을 다루고 있지만, 다양한 출력에 대해서는 해결책이 제시되어 있지 않다. 마지막으로, 2LFC는 각각의 모듈에 퍼지 제어를 갖기 때문에 전형적인 인공 신경망과는 차이가 있다. 오히려 퍼지이론과 신경망 이론이 통합된 퍼지-뉴로 모델에 가깝다고 할 수 있다. 하지만, 2LFC는 퍼지 룰에 큰 비중을 두고 있기 때문에 각 모듈이 연결되는 가중치는 고려하지 않는다.

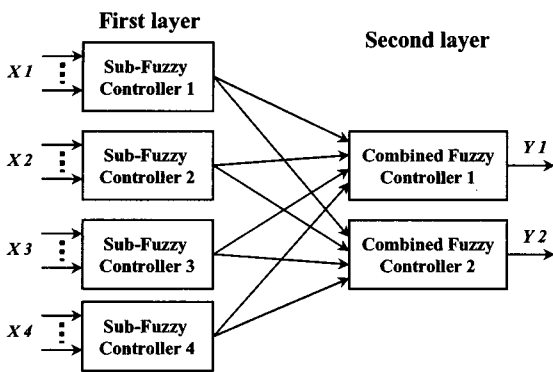


그림 3. 2-layer fuzzy controller의 구조: 이 예는 4개의 sub-controllers와 2개의 combined controllers를 가진다.
Fig. 3. Configuration of the 2-layer fuzzy controller: this example has 4 sub-controllers and 2 combined controller.

2.2. 2-Layer Fuzzy Controller의 적용

본 논문에서는 2LFC의 실제적인 적용을 위해 로봇의 두 가지 행동을 위해 2LFC를 구성하였다. 하나는 이동 로봇의 가장 기본적인 행동으로 장애물 회피를 위해 2LFC를 구성하였고, 다른 하나는 로봇이 다른 로봇을 따라가는 행동을 2LFC로 구성하였다. 그림 4는 이러한 로봇을 위한 2LFC의

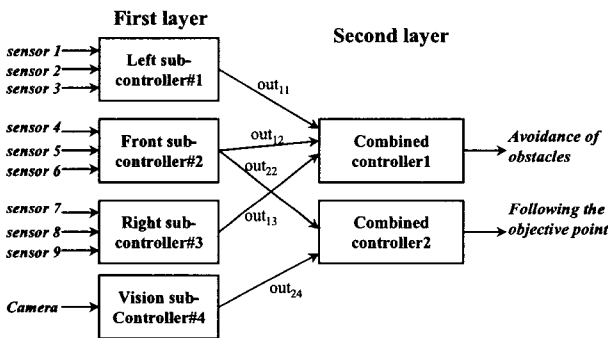


그림 4. 2-layer fuzzy controller의 적용 예: 여러 개의 센서를 입력으로 하여 로봇의 장애물 회피 및 목표물 따라가기 행동을 제어.
Fig. 4. Application of the 2-layer fuzzy controller: avoidance of obstacles and object following control for the robot with many sensors as inputs

구조를 보여준다. 장애물 회피를 위해서는 combined controller1이 사용되고 vision controller와 combined controller2는 사용되지 않으며, 목표물을 따라가는 행동을 위해서는 전체 제어가 모두 사용된다. 주변의 물체의 거리를 알아내기 위해 초음파 센서가 사용되는데, 그 개수가 9개 이므로 이것을 각각 3개씩 분류하여 각각의 sub-controllers로 입력된다. 로봇에 장착된 카메라를 통해 얻은 영상 정보는 vision controller로 입력되어 목표물 탐색에 사용된다.

Vision controller에 대해서는 로봇 축구에서 사용되는 알고리즘과 유사한 방식을 사용하여 단순히 컬러 인식만으로 물체를 탐색하도록 하였으며, 1차원 영상 히스토그램 추출과정을 추가하여 그 효율성을 높였다 [8].

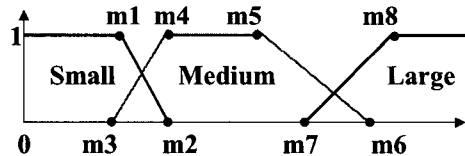


그림 5. 적용에서 사용되는 멤버쉽 함수
Fig. 5. Membership functions in the application

본 논문에서는 각각의 퍼지 제어기에서 그림 5와 같이 세 개의 멤버쉽 함수를 갖도록 설계한다. 또한 퍼지 추론 엔진에서는 TSK model의 simplified method를 사용한다. 이 방식의 장점은 후건부가 상수로 주어지며, 퍼지 디코더를 포함하고 있어서 전체 퍼지 제어기의 구조가 간단해진다는 것이다. Simplified method를 사용했을 경우, 퍼지 제어기의 최종 출력은 다음 식과 같이 주어진다. 여기에서, λ_i 는 i 번째 퍼지 룰의 적합도를 나타내고 c_i 는 i 번째 퍼지 룰의 후건부 상수 값을 가리키며, Z 는 최종 출력 값을 의미한다 [1].

$$Z = \left(\sum_{i=1}^n (\lambda_i \times c_i) \right) / \left(\sum_{i=1}^n \lambda_i \right) \quad (1)$$

세 개의 입력을 갖는 퍼지 제어기에서 각각의 퍼지 룰은 다음과 같은 형태를 갖는다:

if $X1$ is $A1$ and $X2$ is $A2$ and $X3$ is $A3$ then Y is c

$X=(X1, X2, X3)$ 와 Y 는 언어 변수이고 ($A1, A2, A3$)는 멤버쉽 함수를, c 는 후건부 상수 값을 나타낸다. 그림 5에서 알 수 있듯이, 세 개의 멤버쉽 함수는 8개의 파라미터가 필요하다; $m1 \sim m8$. 세 개의 입력에 대해 세 개의 멤버쉽 함수를 가지므로, 각각의 퍼지 제어기는 27개의 퍼지 룰을 가지며, 마찬가지로 27개의 후건부 상수 c 가 필요하다. $c1 \sim c27$. 본 적용에서는 구현을 간단히 하기 위해, 후건부 상수 값을 로봇의 방향 회전 값으로 할당하였다. 즉 로봇은 일정한 속도로 움직이며, 방향 회전 만으로 장애물을 회피하거나 로봇을 따라가게 된다.

이러한 구조를 갖는 퍼지 제어기에 대해 일반적인 퍼지 제어기와 비교해 보면, 9개의 입력과 3개의 멤버쉽 함수를 가지므로 일반적인 퍼지 제어기에서는 $3^9 = 19,683$ 개의 퍼지 룰이 필요하다. 반면에, 2LFC는 각각의 sub-controller에서 $3^3 = 27$ 개씩, combined controller에서도 마찬가지로 27개의 퍼지 룰이 필요하므로 전체 시스템에 대해서는 $27 \times 4 = 108$ 개의 퍼지 룰이 필요하다. 즉, 일반적인 퍼지 제어기가 2LFC보다 180배가 넘는 퍼지 룰을 필요로 한다.

3. SCEA의 적용

3.1 Schema Co-Evolutionary Algorithm

다른 공진화 알고리즘처럼, SCEA도 두 가지의 서로 협조적으로 작용하는 개체군을 갖는다: 숙주-개체군과 기생 개체군. 숙주-개체군은 찾고자 하는 해결책의 후보들로 구성되고 전형적인 유전자 알고리즘과 같은 방식으로 진화하고, 기생-개체군은 유용한 스키마를 찾기 위한 스키마타의 집합으로 구성된다. 그림 6은 SCEA 대략적인 알고리즘을 보여준다 [5].

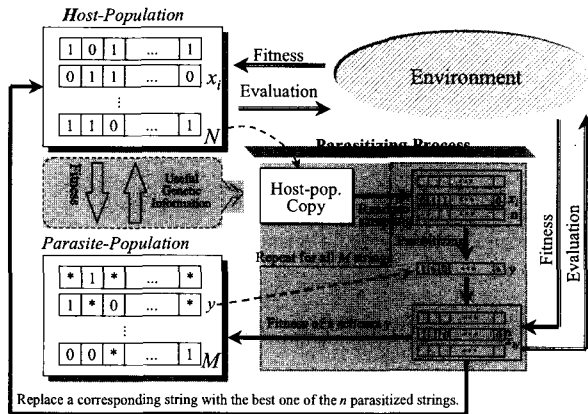


그림 6. 스키마 공진화 알고리즘의 개략도
Fig. 6. Outline of the schema co-evolutionary algorithm

SGA(Simple Genetic Algorithm)은 한 세대동안 네 가지 중요한 단계를 갖는데, 그것은 적합도 평가, 선택, 교차, 돌연변이이다. 그러나 SCEA는 선택과정 전에 기생 프로세스를 추가로 포함한다. 숙주-개체군의 모든 개체들이 평가되어진 후에, 그 중의 몇 개가 기생-개체군의 스키마를 위해 임의로 선택된 다음, 기생 프로세스가 수행된다. 이것을 통해 새롭게 생성된 개체를 평가하고, 원래의 개체와 기생 프로세스에 의해 변한 개체 사이의 적합도 개선 정도를 측정한다. 가장 큰 개선 값을 갖는 기생 개체를 각각의 스키마에 대응하는 개체와 대치한다. 개선된 양을 이용하여, 기생-개체군에서 각각의 스키마의 적합도를 부여할 수 있다. 그러므로 기생-개체군의 스키마의 적합도는 스키마의 유용성을 가리킨다. 적합도 평가 후의 기생-개체군에는 SGA와 동일한 과정을 적용한다. 기생-개체군의 개체의 적합도는 기생 프로세스에 의해 계산된다. 따라서 기생-개체군은 기생, 선택, 교차, 돌연변이의 네 단계를 거친다. 기생 프로세스는 [5]에 자세히 나타나 있다.

3.2 SCEA를 이용한 2LFC의 최적 설계

2.2 절에서 구성한 2LFC에서는 각각의 퍼지 제어기에 대해 전진부 멤버쉽 함수를 위해 8 개, 후진부 상수를 위해 27 개의 파라미터가 필요하므로 총 35개의 파라미터가 결정되어야 한다. 하나의 파라미터를 6비트로 잡으면 거리는 64cm 까지를 범위로 하고 로봇의 방향 회전 값은 $\pm 10^\circ$ 를 오차로 했을 경우에는 충분히 전방향에 대해 제어가 가능하다. 따라서 진화 연산을 적용하기 위해 필요한 비트 스트링은 210비트가 된다. 즉, SCEA와 SGA에서 모두 210비트 길이를 갖는 개체를 진화시킨다. 그림 7은 이러한 개체 스트링을 보여준다.

| | | | | | | | |
|-----------|-----------|-----|-----------|-----------|-----------|-----|------------|
| <i>m1</i> | <i>m2</i> | ... | <i>m8</i> | <i>c1</i> | <i>c2</i> | ... | <i>c27</i> |
| 6bits | 6bits | | 6bits | 6bits | 6bits | | 6bits |
| 48bits | | | | 162bits | | | |
| 210bits | | | | | | | |

그림 7. SCEA를 적용하기 위한 개체의 비트 스트링
Fig. 7. Bit string of an individual in the application of the SCEA

2LFC에서, combined controller가 통합적인 추론을 하고 최종 출력을 생성하므로 sub-controllers 보다 더 중요하다. 따라서 combined controller의 퍼지 물이 먼저 결정되고, 그 다음 sub-controllers의 퍼지 물이 차례로 결정된다. 다음에 SCEA를 이용한 2LFC의 최적 설계 단계를 요약하여 나타내었다.

- Step 1 :** First of all, the SCEA designs the combined controller with 3 inputs.
- Step 2 :** After adding a left sub-controller with 3 inputs, the SCEA designs this sub-controller.
- Step 3 :** After adding a right sub-controller with 3 inputs, it is designed.
- Step 4 :** After adding a front sub-controller with 3 inputs, it is designed by the SCEA.

목표물 탐색을 위한 비전 시스템은 SCEA에 의해 결정되어지지는 않으나 SCEA가 수행되는 동안 시스템에 추가되어 목표물의 위치를 계속적으로 알려줌으로써 로봇 제어를 위한 퍼지 물 생성에 영향을 미치도록 하였다.

진화 연산에서 또 하나 중요한 것은 적합도 함수를 설정하는 것이고, 이것은 적용하는 문제에 따라 달라진다. 본 연구에서는 다음 식과 같은 적합도 함수를 사용하였다.

$$f = (pos/11) \times (1 - close/50) \times 15/time \quad (2)$$

로봇이 움직일 거리를 11 부분으로 나누고 시작점은 0점을 도착점은 11점을 주도록 하였다. 'pos'는 이러한 로봇의 움직인 거리를 나타낸다. 또한 로봇이 벽에 너무 가까이 다가가지 않도록 하기 위해 'close'항을 추가하였다. 즉 로봇이 장애물에 5cm 이내로 접근한 횟수를 측정하여 'close'로 값을 준다. 즉 장애물이 가까이 가지 않은 로봇은 'close'값이 0이 되어 높은 적합도를 갖게 된다. 또한 총 주행시간을 적합도 함수에 반영하여 로봇이 장애물에 가까이 가지 않고 도착점까지 갔더라도 더 빨리 주행한 로봇에게 높은 적합도를 부여한다. 물론 로봇이 끝까지 주행하지 않은 경우에는 'time'은 가장 큰 값(20)을 주어 도착점에 가지 않고 이미 주행이 끝난 로봇에 대해서는 낮은 적합도를 부여하도록 하였다. 여기서 15는 전체 적합도의 최대치를 1로 맞추기 위한 스케일링 값이다. 결과적으로, 장애물에 너무 가까이 가지 않으면서 빠른 시간에 도착점까지 충돌 없이 주행한 로봇이 최고의 적합도를 부여받는다. 물론 장애물의 충돌이 있을 경우에는 로봇의 주행이 바로 정지되며, 적합도는 0을 받는다.

4. 실험 및 결과

4.1 SCEA와 SGA의 비교

앞에서 언급한 SCEA와 단순한 유전자 알고리즘(SGA)을 비교하기 위해, 같은 퍼지 파라미터를 갖는 combined

controller에 대해 컴퓨터 시뮬레이션을 수행하였다. 즉 3.2절에서 설명한 SCEA 과정의 Step 1에 대해서만 시뮬레이션을 하였다. 이때 SGA의 개체군의 크기는 100으로 하고, SCEA의 숙주-개체군은 40, 기생-개체군은 30, 기생 프로세스를 위한 샘플링 크기는 2로 하여, SGA와 SCEA의 각 세대에서의 평가 횟수가 같도록 설정하였다; $100 = 40 + (30 \times 2)$.

각각에 대해 100번의 시뮬레이션을 수행하였고, 엘리트 보존 선택이 사용되었다. 각각의 진화는 150세대까지만 수행되었고, 기생-개체군의 교차 확률과 돌연변이 확률은 각각 0.05과 0.1로 고정시켰다. 숙주-개체군 또는 SGA의 개체군에서의 교차 확률과 돌연변이 확률은 표 1에 나타나 있는 것과 같이 다양하게 조정하면서 시뮬레이션을 하였다. 표 1에서 최적의 결과는 적합도가 0.95이상인 경우를 나타낸다.

표 1에서 알 수 있듯이 SCEA가 SGA보다 최적해를 더 잘 찾는 것을 알 수 있다. 특히 SCEA는 파라미터의 변화에 상관없이 100회 중에 거의 90회 이상 최적해를 찾아냈고, 최적해를 찾을 확률은 96.4%가 되어 높은 최적해 탐색률을 보여준다. 반면에 SGA는 파라미터의 변화에 민감한 편이며, 76%의 최적해 탐색률을 보여준다. 또한 최적 결과 달성 세대 평균에서도 알 수 있듯이, 대체로 SCEA가 SGA보다 더 빨리 최적해를 찾는 것을 볼 수 있다. 표 1에서 간혹 SGA의 탐색 세대수가 더 적은 경우가 있는데, 이것은 최적해를 탐색하지 못한 경우를 제외했기 때문이다.

4.2 SCEA를 이용한 최적의 2LFC 설계

SCEA의 Step 1, 2, 3, 4를 거쳐 최적의 2LFC를 설계하였다. SCEA는 파라미터의 변화에 덜 민감하므로 교차확률 0.5와 돌연변이 확률 0.001, 기생-개체군의 교차확률 0.05, 돌연변이 확률 0.1로 설정하고 최적의 퍼지 룰을 찾아가도록 하였다. SCEA의 각 단계에서는 모두 최대 적합도가 1.0이 될 때까지 퍼지 룰을 진화시켰다. 각각의 퍼지 제어기에 대해 얻어진 멤버십 함수를 그림 8에 나타내었다. Step 1에서는 41세대에 최적의 퍼지 룰을 찾아냈다. Step 2는 18세대, Step 3는 8세대, Step 4는 5세대에서 최적의 퍼지 룰을 찾아냈다.

표 1. SCEA와 SGA를 이용하여 최적의 결과(최대 적합도가 0.95이상)를 찾았을 경우의 횟수 및 달성한 평균 세대. (GA 파라미터 P_c 와 P_m 을 바꿔 가면서 150세대까지 100번씩 수행한 결과)

Table 1. The number of best results and average generation of best results in 100 runs.

| Pc | Pm | best result times | | best result avg. generation | |
|-----|-------|-------------------|------|-----------------------------|-------|
| | | SGA | SCEA | SGA | SCEA |
| 0.3 | 0.001 | 48 | 97 | 50.83 | 30.42 |
| | 0.01 | 82 | 99 | 30.1 | 31.6 |
| | 0.1 | 95 | 97 | 46.87 | 40.98 |
| 0.5 | 0.001 | 52 | 87 | 23.60 | 32.82 |
| | 0.01 | 89 | 99 | 26.88 | 24.93 |
| | 0.1 | 92 | 96 | 46.74 | 47.49 |
| 0.7 | 0.001 | 53 | 93 | 31.23 | 31.17 |
| | 0.01 | 91 | 100 | 19.82 | 23.31 |
| | 0.1 | 82 | 100 | 57.74 | 49.27 |

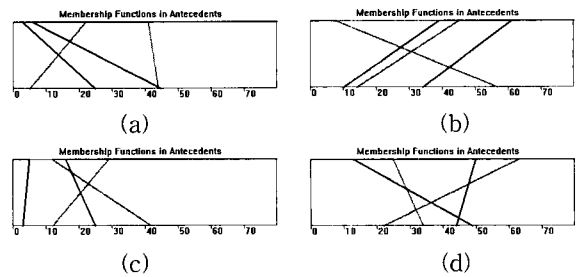


그림 8. 2LFC에서 각 퍼지 제어기의 멤버십 함수 (a) step1에 의해 찾은 combined controller의 전건부 멤버십 함수 (b) step2, left sub-controller (c) step3, front sub-controller (d) step4, right sub-controller
Fig. 8. Membership functions of each fuzzy controller in 2LFC (a) in step1, combined controller (b) in step2, left sub-controller (c) in step3, front sub-controller (d) in step4, right sub-controller

4.3 로봇 주행 실험

위의 시뮬레이션을 통해 찾은 2LFC를 그대로 실제 로봇에 적용하여 주행 실험을 하였다. 그림 9는 로봇의 장애물 회피에 관한 실험 모습이고, 그림 10은 로봇 따라가기에 대한 실험 모습이다. 그림 9에서는 시뮬레이션에서와 마찬가지로 장애물 회피뿐만 아니라 벽을 따라가는 것을 볼 수 있다. 이것은 가장 빠른 길을 찾는 과정에서 SCEA가 스스로 벽을 따라가는 것이 효율적이라고 판단하여 2LFC를 그러한 구조로 진화시킨 결과로 보인다. 로봇 따라가기에서는 앞에 가는 로봇은 장애물까지의 거리가 100cm 이상인 경우에는 미리 정의된 경로를 따라 움직이도록 하였고, 그 이하인 경우에만 2LFC가 작동하도록 하였다. 뒤에서 따라가는 로봇은 앞의 시뮬레이션에서 비전 컨트롤러를 추가한 후에 수행한 결과로 생성된 2LFC를 이용하여 주행하도록 하였다.



그림 9. 로봇의 장애물 회피 실험.
Fig. 9. Experiment on avoidance of obstacles

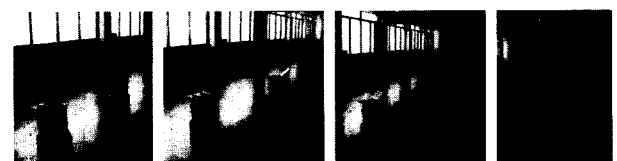


그림 10. 로봇 따라가기
Fig. 10. Experiment on robot following

5. 결론

본 논문에서는 다양한 입력을 이용하여 다양한 행동을 출력하면서도 퍼지 룰의 개수가 적은 2-layer fuzzy controller를 사용하여 이동 로봇을 제어하였다. 로봇은 많은 초음파 센서와 비전 카메라를 장착하고 있으며, 본 논문에서는 장애물 회피와 벽 따라가기, 로봇 따라가기 등을 수행하였다. 퍼

지 제어기의 가장 중요한 문제점은 퍼지 룰을 어떻게 설계하느냐에 달려 있는데, 이를 위해 schema co-evolutionary algorithm을 적용하였다. SCEA를 이용하여 최적의 2LFC를 설계하였으며, 실제 실험을 통해 그 효율성을 입증하였다.

2LFC는 복잡한 퍼지 제어기를 몇 개의 sub-controllers와 combined controller로 나누어 처리하므로 전체 시스템의 구조를 나타내기 쉽고, 적은 개수의 퍼지 룰을 이용하므로 매우 효율적인 알고리즘이다. 또한 SGA보다 속도 및 성능 면에서 우수한 SCEA를 적용하여 최적해를 빠르고 쉽게 찾을 수 있다. 그러므로, 지금까지 또는 앞으로 직면할 구현하기 어려운 제어기나 그 설계 및 최적해 탐색에서, 2-layer fuzzy controller와 schema co-evolutionary algorithm이 다양한 분야에 적용되어질 것으로 기대된다.

참 고 문 헌

- [1] 강훈, 심귀보, *지능 정보 시스템*, 대영사, 2001.
- [2] 심귀보, 변광섭, 박창현, "이동로봇의 행동제어를 위한 2-Layer Fuzzy Controller," *한국퍼지 및 지능시스템 학회 논문지*, 제13권 3호, pp. 287-292, 2003.
- [3] L. A. Zadeh, "Soft computing and fuzzy logic," *Journal of IEEE on Software*, vol. 11, Issue 6, pp. 48-56, 1994.
- [4] 심귀보, 전호병, "스키마 공진화 알고리즘의 해석," *한국퍼지 및 지능시스템학회 논문지*, 제10권 1호, pp. 23-27, 2000.
- [5] K.B. Sim, H.B. Chun, and D.W. Lee, "Dynamic behavior control of autonomous mobile robots using schema co-evolutionary algorithm," *Proceedings of IEEE International Symposium on Industrial Electronics (ISIE 2001)*, vol. 1, pp. 560-565, 2001.
- [6] F.L. Chung and J.C. Duan, "On multistage fuzzy neural network modeling," *IEEE Transaction on Fuzzy Systems*, vol. 8, no. 2, pp. 125-142, 2000.
- [7] J.C. Duan and F.L. Chung, "Multilevel fuzzy relational systems: structure and identification," *Soft Computing 6*, Springer-Verlag 2002, pp. 71-86, 2002.
- [8] 김종환 외 8명, *로봇 추구 공학*, KAIST Press, 2002.

저 자 소 개



심귀보(Kwee-Bo Sim)

1984년 : 중앙대학교 전자공학과 공학사
 1986년 : 동대학원 전자공학과 공학석사
 1990년 : The University of Tokyo
 전자공학과 공학박사
 1991년 ~ 현재 : 중앙대학교 전자전기
 공학부 교수

2000년 ~ 현재 : 제어자동화시스템공학회 이사 및 지능
 시스템연구회 회장
 2002년 ~ 현재 : 중앙대학교 산학연컨소시엄센터 센터장
 2003년 ~ 현재 : 일본계측자동제어학회(SICE) 이사
 2003년 ~ 현재 : 한국퍼지 및 지능시스템학회 부회장

관심분야 : Artificial Life, Intelligent Robot, Intelligent
 Systems, Artificial Brain, Multi-Agent System, Distributed
 Autonomous Robotic System, Machine Learning, and
 Adaptation Algorithm, Soft Computing(Neuro, Fuzzy,
 Evolutionary Computation), Evolvable Hardware, Artificial
 Immune System, Intrusion Detection System, etc.

Phone : +82-2-820-5319

Fax : +82-2-817-0553

E-mail : kbsim@cau.ac.kr

Homepage URL : <http://alife.cau.ac.kr>



변광섭(Kwang-Sub Byun)

2003년 : 중앙대학교 전자전기공학부
 공학사

2003년~현재 : 동대학원 전자전기공학부
 석사과정

관심분야 : 퍼지 제어기 설계, 로봇 원격제어, 공진화 알고리즘

Phone : +82-2-820-5319,

E-mail : ks4070@yahoo.co.kr