
유전자 알고리즘을 이용한 자동차 주행 제어기의 최적화

김봉기

Optimization of Simulated Fuzzy Car Controller Using Genetic Algorithm

Bong-Gi Kim

요 약

퍼지 논리 제어기(FLC : Fuzzy Logic Controller)를 사용할 때, 가장 중요한 것은 소속 함수의 범위를 정하는 것과 규칙의 형태를 결정하는 것이다. 소속 함수의 범위나 규칙의 형태는 지금까지 전문가가 임의로 정하는 방법을 사용하였다. 그러나 기존의 방법을 사용하면, 전문가의 주관적인 규칙과 소속 함수가 생성될 수 있고, 소속함수의 경우 최적의 범위를 정확히 예측하기 어려운 단점이 있다. 본 논문에서는 이런 단점을 보완하기 위해, 유전자 알고리즘을 사용함으로써 최적의 소속 함수와 규칙의 형태를 구하려 하였다. 제시하는 방법의 타당성을 검증하기 위해 자동차 주행 제어 문제에 적용시켜 보았다.

ABSTRACT

The important problem in designing a Fuzzy Logic Controller(FLC) is generation of fuzzy control rules and it is usually the case that they are given by human experts of the problem domain. However, it is difficult to find an well-trained expert to any given problem. In this paper, I describes an application of genetic algorithm, a well-known global search algorithm to automatic generation of fuzzy control rules for FLC design. Fuzzy rules are automatically generated by evolving initially given fuzzy rules and membership functions associated fuzzy linguistic terms. Using genetic algorithm efficient fuzzy rules can be generated without any prior knowledge about the domain problem. In addition expert knowledge can be easily incorporated into rule generation for performance enhancement. We experimented genetic algorithm with a non-trivial vehicle controlling problem. Our experimental results showed that genetic algorithm is efficient for designing any complex control system and the resulting system is robust.

키워드

Genetic algorithm, Fuzzy car controller, Optimization

I. 서 론

퍼지 논리 제어기란 퍼지 규칙과 소속 함수를 이용하여 대상이 되는 시스템을 제어하는 것을 말한다. 기존에는 퍼지 규칙과 소속 함수를 전문가가 직접 설계하는 방식을 사용하였다. 기존의 방법에서 전문가가 제시한 규칙이나 소속 함수가 최적이라는 보장이 없으며, 대부분의

사람이 그러하듯이 주관적인 생각이 많이 개입될 수 있어서 객관적이지 못하다. 이런 문제들 때문에 최적의 규칙과 소속 함수를 찾기 위하여 신경망이나, 유전자 알고리즘을 사용한 여러 가지 방법들이 제어되어 왔다. 본 논문에서는 자동차주행 제어 문제에 유전자 알고리즘을 적용하여 퍼지 규칙과 소속 함수의 최적화를 시도하였다.

II. 관련 연구

퍼지 논리 제어란 인간 전문가의 프로세서에 대한 지식과 경험에 기초한 전문지식을 언어적 제어규칙으로 표현하고 이의 추론과정을 통하여 제어기를 동작시키는 시스템을 말한다. 이는 수학적으로 시스템을 표현하기가 복잡하거나 어려운 경우 사용되며, 복잡한 수학적 모델에 의존하지 않고도 전문가의 직감이나 경험상의 법칙, 실제 실험 또는 모의실험의 데이터를 통해서 제어에 필요한 규칙들을 유도해 낼 수가 있다. 퍼지 제어시스템은(그림 1)에서 보이는 바와 같이 퍼지화부, 비퍼지화부, 퍼지추론부 및 지식 베이스의 네 가지 구성 요소로 되어 있다[1].

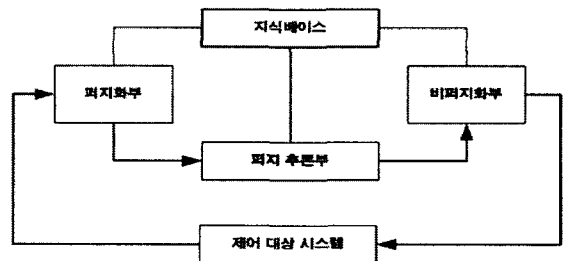
- ① **퍼지화부(Fuzzifier)**는 센서로부터 입력된 값을 전체 집합상의 값들로 퍼지 사상시키기 위하여 측정된 값을 퍼지화하는 함수를 이용하여 적당한 언어 값으로 변환한다.
- ② **지식 베이스(Knowledge Base)**는 If~then 형식의 퍼지 규칙과 각 입출력 변수에 적용되는 퍼지소속함수를 지식 베이스로 구축한다. 퍼지제어에서는 이 부분이 가장 중요하며, 전체 시스템의 안정도 및 강인성이 이 퍼지제어규칙의 설계 방법에 달려 있다. 가장 쉽게 설계하는 방법은 전문가의 경험과 지식을 동원하여 직관적인 규칙을 만드는 것이다.
- ③ **퍼지 추론부(Fuzzy Inference Engine)**는 퍼지 논리 제어기의 실행부로, 구축된 지식 베이스를 참조하면서 퍼지화된 입력값을 추론의 합성규칙을 이용하여 결론적 퍼지 집합으로 추론한다.
- ④ **비퍼지화부(Defuzzifier)**는 출력 변수에 할당된 값을 전체 집합위의 값들로 퍼지 사상시키기 위하여 추론된 퍼지 제어값을 하나의 값으로 비퍼지화한다. 비퍼지화 방법으로는 면적 중심 방법, 최대 표준 방법, 최대 평균 방법 등이 있다.

퍼지 제어의 기본이 되는 퍼지 논리(Fuzzy Logic)는 불확실한 인간의 근사적 추론 방법을 모델링하기 위한 논리적 체계로서 인간 판단 등의 애매성을 포함하는 불확실한 영역에 대해 유연하고 강인한 지능적 추론을 할 수 있다. 이러한 퍼지 논리는 기존의 논리 체계에 비해 인간의 사고나 자연어의 특성과 많은 유사성을 지니고 있어서, 시스템의 정보가 근사적이고 불확실한 시스템의 제

어나 패턴인식 등과 같은 문제에 많이 응용되고 있다[2].

이러한 퍼지 논리 제어기를 설계하기 위해서, 기존에는 퍼지 규칙과 소속 함수를 전문가가 직접 설계하는 방식을 사용하였다. 그러나 원하는 분야의 문제 영역의 많은 문제들을 접해 본 전문가를 찾기 어려우며, 전문가가 없이 제어기를 설계한다는 것은, 불충분하고, 비효율적인 규칙들을 생성할 뿐만아니라, FLC의 적응력 부족으로 인한 시행착오를 통한 시간과 비용의 낭비 등의 문제를 초래한다. 그리고 전문가를 찾았더라도 일반적으로 사람들은 객관적인 생각보다 주관적인 생각을 많이 개입시키기 때문에, 전문가가 정의한 규칙이나 소속 함수가 최적이라는 보장이 없다. 또한 제어 대상 시스템이 바뀔 때마다 제어기를 다시 정의하기가 곤란하다. 이러한 문제들을 해결하기 위한 새로운 퍼지 시스템의 설계 방법들로 신경망을 이용하여 퍼지 제어 규칙을 학습하는 방법[3][4], 진화 알고리즘을 이용한 방법[5][6][7][8][9] 등이 제안되었다.

본 논문에서는 유전자 알고리즘을 사용하여 퍼지 제어 시스템을 설계한 것에 대하여 기술한다. 유전자 알고리즘을 사용하면 문제에 대한 지식이 없이도 제어기를 설계할 수 있으며, 만약 문제에 대한 지식이 있다면 그 지식을 이용하여 진화 효율을 향상시킬 수 있고, 신속하면서도 전문가가 정의한 규칙보다 더 좋은 결과를 낼 수 있다. 또한 제어대상 시스템의 환경이나 동작 원리가 바뀌어도 변화에 신속하게 대응하여 제어기를 설계할 수 있는 장점이 있다.

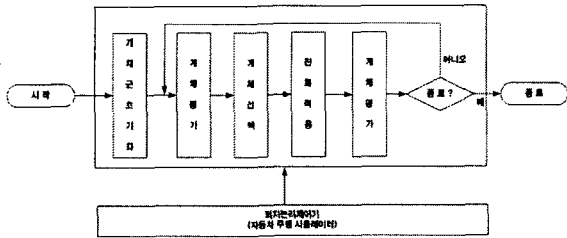


(그림 1) 퍼지 제어 시스템의 기본 구조
Fig.1 Fundamental Structure of Fuzzy Control System

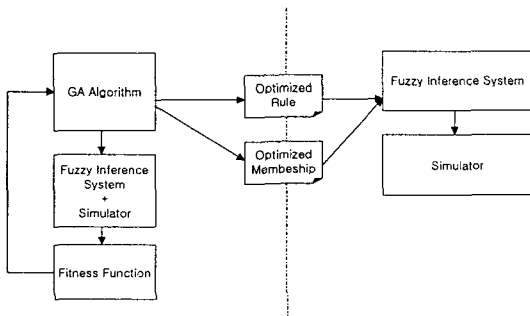
III. 자동차 주행 제어 시스템 구성

본 논문에서 제안하는 시스템은(그림 2)와 같이 진화

알고리즘의 틀에 의해 퍼지 논리 제어를 진화시키도록 구성되어 있다. 먼저 주행 제어문제에 필요한 매개변수들을 정하고, 그 매개 변수들을 이용하여 염색체의 형태를 설계한 뒤에 가능한 해(자동차 주행 제어기)를 만들기 위하여 매개변수에 해당되는 값들을 무작위로 초기화한다. 그리고 자동차 주행 시뮬레이터를 이용해 주행 시험을 하여 제어기(염색체)들을 평가한 후, 진화 루프(loop)를 통해 최적의 염색체로 진화시킨다. (그림3)은 유전자 알고리즘을 사용하여 최적화하여 얻은 규칙과 소속 함수를 퍼지 제어기에 사용하는 전체 시스템 구성도이다.



(그림 2) 자동차 주행제어 시스템
Fig.2 Car Driving Control System



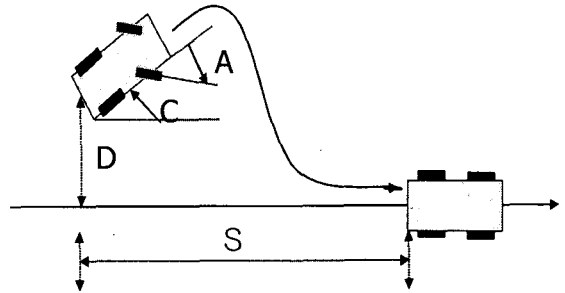
(그림 3) 최적화 수행 구조
Fig.3 Optimized Achievement Structure

IV. 퍼지 논리 제어를 이용한 자동차 주행 제어

본 논문에서 제안하는 자동차 주행제어기는 앞에서 설명한 퍼지 논리 제어를 사용하도록 하였다. 자동차 주행 시험을 시뮬레이션하기 위해서 필요한 자동차와 도로를 정의하고, 퍼지 논리 제어기가 주어진 시뮬레이션 상황을 제어하였다.

4.1 자동차의 정의

실제적인 자동차를 시뮬레이션하는 것은 구현상 무리가 있다. 그래서 본 논문에서는 자동차를 하나의 점의 형태로 정의하고, 자동차의 방향, 속도, 바퀴의 방향 등의 정보를 저장하여 사용하였다. (그림 4)와 같이 자동차의 주행에 필요한 정보를 자료구조에 형태로 저장하였다.

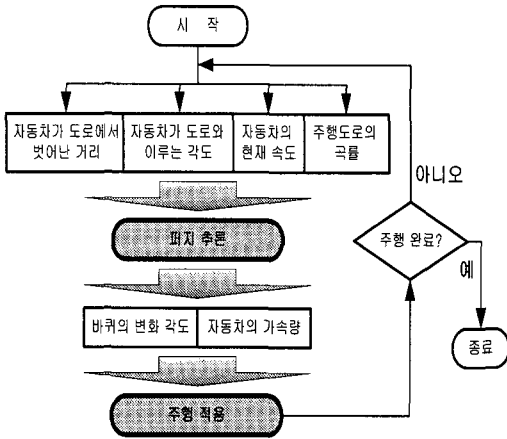


(그림 4) 자동차의 모델링
Fig.4 Car Modeling

4.2 도로의 정의

본 논문에서 사용되는 도로데이터는 직선도로, 굴절도로, 회전도로, 복합도로의 다양한 형태로 정의하였다. 도로데이터를 정의하기 위해 실제도로의 정보를 가능한 손실없이 표현하기 위해 도로를 구성하는 중심점들을 나열하여 저장하는 방식을 사용하였다. 이 방식을 사용하여 직선, 굴절, 회전도로 뿐 아니라 불규칙적인 도로의 변화까지도 반영할 수 있도록 하였다. 한 도로 중심점을 표현하기 위해 X 좌표점, Y 좌표점 그리고 다음 도로 중심점과의 각도의 세 가지 요소를 사용하였다. 도로 중심점의 좌표점의 위치는 도로가 절대 좌표계 상에 있다고 가정하고 각 점들의 절대 위치를 정하였다. 실제 자동차의 이동시 각 점들 간의 상대적인 위치만 의미가 있으며, 절대적인 좌표값의 크기는 자동차 이동에 상관이 없고 단지 도로 데이터를 모델링할 때 사용될 뿐이다. 구분 단위를 정함에 있어 실제도로에서 1미터 간격에서는 큰 변화 즉, 각도의 갑작스런 변화가 나타날 수 없는 점을 감안해 1미터를 최소단위로 정하였다.

예를 들어 아래 (그림 5)와 같이 직선도로에서 45도 꺾어진 대각선 도로가 나타난다고 할 때 표현은 다음과 같다. 실제 위 데이터는 텍스트 파일에 (0, 0, 0; 1, 0, 0; 2, 0, 45, 3, 1, ?)의 형식으로 정의되어 있다. 위 도로데이터는 2미터간 직선 도로가 나타난 후 45도의 굴절이 나타난 도



(그림 5) 도로의 표현
Fig.5 Road Expressing

로를 의미하고 있다. 마지막점에서는 각도값은 다음 점 이 정의되어 있지 않으므로 각도를 ?로 하였다.

4.3 자동차의 주행 제어

본 논문 사용한 자동차 주행 제어 시스템은 가변속도로 움직이는 자동차가 주어진 주행선(주행도로의 중앙선)을 따라 가능한 한 최단시간에 주행하도록 자동차 핸들(바퀴)을 제어하는 문제이다. 퍼지 논리 제어기는 자동차가 도로에서 떨어진 거리, 자동차가 도로와 이루는 각도, 자동차의 속도 그리고 주행도로의 곡률을 자동차의 센서로부터 입력받아서, 바퀴의 변화 각도와 자동차의 가속량을 출력한다.

본 논문에서는 도로에서의 자동차 주행 제어를 위하여 (그림 6)와 같은 방법으로 수행하였다.

Distance = { Left, Middle, Right }, [-120, 120]
 Angle = { Left, Middle, Right }, [-60, 60]
 Speed = { Low, High }, [0, 75]
 Curve = { Left, Middle, Right }, [-180, 180]

입력변수

(그림 6) 도로 주행 제어 모델
Fig.6 Car Driving Control Model

퍼지 논리 제어기에서 사용한 입력 변수는 자동차가 도로에서 떨어진 거리(Distance)와 자동차가 주행할 도로

와 이루는 각도(Angle), 자동차의 속도(Speed) 그리고 주행도로의 곡률(Curve)이 있으며, 출력 변수는 자동차의 바퀴가 변화해야 할 각도(Steering)와 자동차의 가속량(Acceleration)이다. 각 변수에서 사용되는 언어 항들의 집합과 그 각각의 범위는 (그림 7)과 같다.

Steering = { Left, Middle, Right }, [-60, 60]
 Acceleration = { Negative, Zero, Positive }, [-50, 50]

출력변수

(그림 7) 입력 변수와 출력 변수의 값
Fig.7 Value of Input/Output Variables

(그림 7)에서 거리(distance)변수는 주행선상의 주행 방향에서 자동차가 도로의 왼쪽에 있으면 음의 값을, 오른쪽에 있으면 양의 값을 갖도록 하였다. 각도(angle), 곡률(curve), 바퀴각도(steering)변수도 마찬가지로 주행선상의 주행방향을 0이라고 했을 때 왼쪽으로 기울어져 있으면 음의 값을, 오른쪽으로 기울어져 있으면 양의 값을 갖도록 하였다. 거리 변수의 단위는 컴퓨터 화면상의 점 사이의 거리를 사용하였고, 각도와 곡률 그리고 바퀴각도 변수의 단위는 도(degree)를 사용하였다. 단위시간을 한 번의 제어가 일어나고 다음 제어가 일어날 때까지로 정의하면서, 속도(speed)는 단위시간동안 진행한 거리를 사용하였고, 가속량(acceleration)은 단위시간동안 속도의 변화량을 사용하였다.

규칙의 개수는 입력변수의 조합에 의해 결정되는데, 본 논문에서 사용한 퍼지 논리 제어기는 4개의 입력 변수에 대해 최대 54개의 규칙을 갖는다.

삽입한다.

V. 유전자 알고리즘의 적용

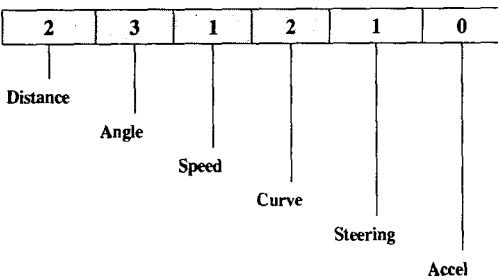
5.1 염색체 구성

5.1.1 규칙과 소속 함수의 염색체

퍼지 논리 제어기의 염색체는 규칙베이스 부분과 소속 함수 부분을 구분하여 인코딩 한다. 이처럼 구분된 인

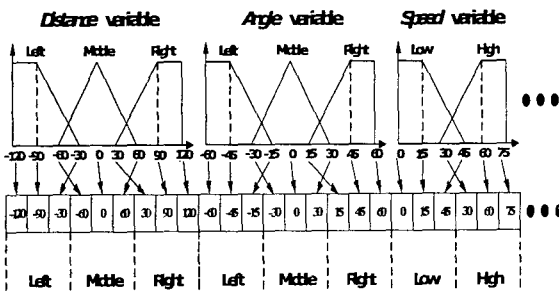
코딩 방법은 규칙과 소속 함수 개체에 각기 특성에 맞는 진화 연산자를 적용할 수 있기 때문에 진화 단계에서 좋은 규칙과 소속 함수의 유지가 쉽다는 점과 인코딩된 개체의 해석이 용이한 점등의 특징이 있다. (그림 8)과 (그림 9)는 규칙 베이스와 소속 함수의 인코딩 형태를 나타낸다. 규칙베이스의 경우는 각 언어항을 정수로 인코딩하였고 소속 함수의 경우는 실수로 인코딩 하였다.

	0	1	2	3
Distance	X	Left	Middle	Right
Angle	X	Left	Middle	Right
Speed	X	Low	High	X
Curve	X	Left	Middle	Right
Steering	X	Left	Middle	Right
Accel	X	Negative	Positive	X



IF Distance is Middle &
 Angle is Right &
 Speed is Low &
 Curve is Middle
 Then
 Steering is Left

(그림 8) 규칙의 인코딩 구조
 Fig.8 Rule Encoding Structure

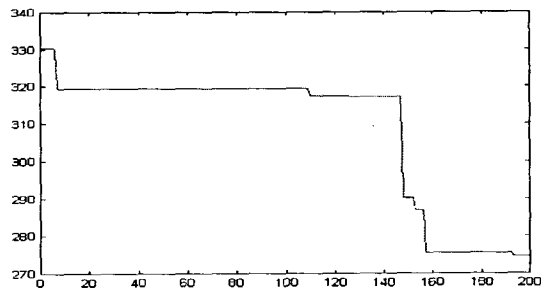


(그림 9) 소속 함수의 인코딩 구조
 Fig.9 Membership Function Encoding Structure

VI. 실험

6.1 규칙의 진화 실험

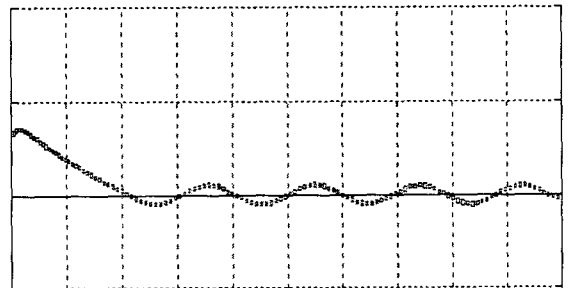
본 실험에서는 GA(Genetic Algorithm)을 사용하여 자동차 제어 규칙을 개선하고자 하였다. 적합도 값은 실제 도로에 주행시켰을 때 나타나는 도로중심점과의 거리와 핸들을 꺾은 횟수 등을 조합하여 계산하였다. 실험환경은 진화 세대는 200세대로 하였으며 한 세대의 개체 수는 20으로 하였다. 아래의 그림은 각 세대별 최고 적합도를 갖는 개체의 값을 보여준다. 이를 통해 진화에 따른 적합도의 Rule의 개선정도를 확인해 볼 수 있다.



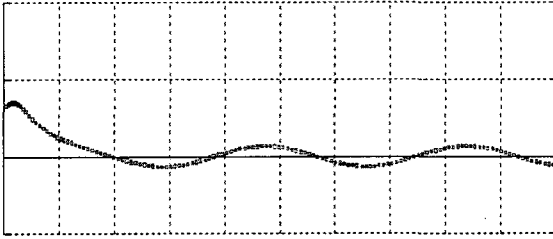
(그림 10) 적합도 진화 과정도
 Fig.10 Fitness Evolution Trend

6.2 주행 실험

본 주행 실험에서는 사람의 정의한 Rule을 사용한 자동차 주행과 GA(Genetic Algorithm)을 사용하여 개선된 Rule을 사용한 주행을 비교하였다.

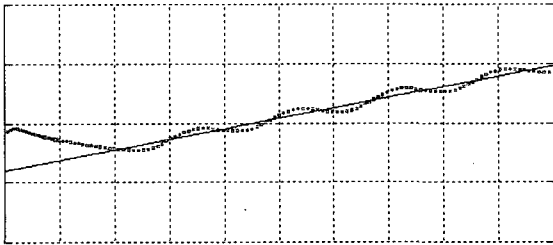


(그림 11) 사람이 정의한 Rule을 사용하여 직선도로를 주행
 Fig.11 Movement of Car Controlled by Man-Defined Rules Along Straight Road



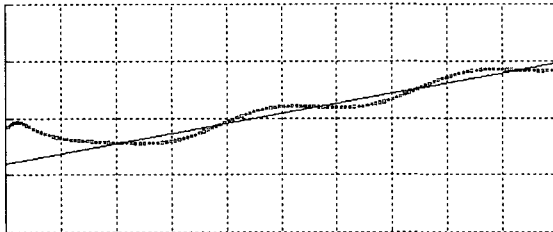
(그림 12) GA로 개선한 Rule을 사용하여 직선도로를 주행

Fig.12 Movement of Car Controlled by GA-Improved Rules Along Straight Road



(그림 13) 사람이 정의한 Rule을 사용하여 대각선 직선도로를 주행

Fig.13 Movement of Car Controlled by Man-Defined Rules Along Cater-Cornered Road



(그림 14) GA로 개선한 Rule을 사용하여 대각선 직선도로를 주행

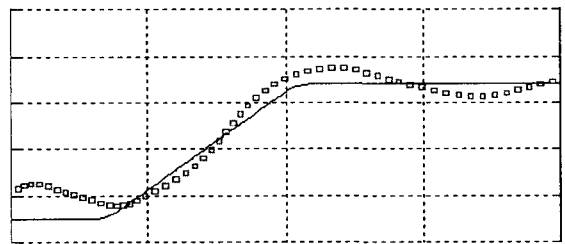
Fig. 14 Movement of Car Controlled by GA-Improved Rules Along Cater-Cornered Road

실험에 적용한 도로 데이터는 여러 환경에서 적응력을 평가하기 위해 직선위주의 단순도로와 직선도로에 회전도로가 복합적으로 나타나는 복합도로 두 가지에 적용해 보았다.

단순 도로에서는 직선 도로가 40미터 정도 나타나다가 약간의 회전 도로가 나타난 후 대각 직선 도로가 나타나며 100미터 부분에서 마지막까지 직선도로를 만나게 된다. 복합도로에는 X축으로 1000미터 길이를 갖는 도로로서 특이할만한 점은 850미터에서 950미터까지 회전 도

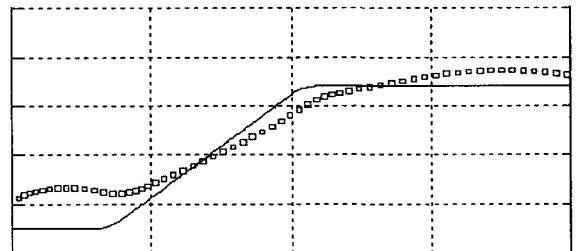
로가 나타나게 되는데 실험 결과에서 어떻게 개선됐는가를 비교해 볼 수 있다.

자동차의 초기 위치 값은 (X:0,Y:20,차바퀴각:60)으로 도로의 중심에서 20미터가 떨어져 있도록 했으며 초기 바퀴의 각도는 도로와 60도가 틀어지도록 하였다. 즉 초기 자동차는 도로의 중심에서 떨어져 있으며 방향 또한 도로와 다른 방향으로 향해 있다. 주행 결과를 통해 도로로 수렴하는 형태를 비교해 볼 수 있을 것이다.



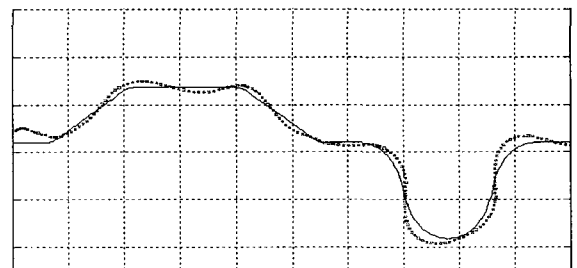
(그림 15) 사람이 정의한 Rule을 사용하여 단순도로를 주행

Fig.15 Car Controlled by Man-Defined Rule along Simple Road Alignment



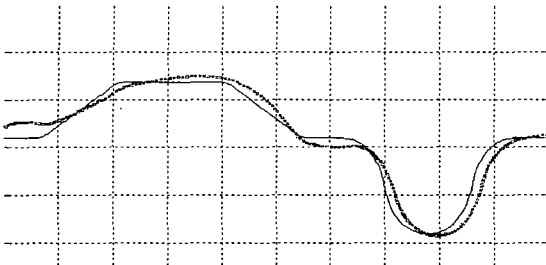
(그림 16) GA로 개선한 Rule을 사용하여 단순도로를 주행

Fig.16 Car Controlled by GA-Improved Rule along Simple Road Alignment



(그림 17) 사람이 정의한 Rule로 복합도로 주행

Fig.17. Car Controlled by Man-Defined Rule along Complex Road Alignment



(그림 18) GA로 개선한 Rule을 사용하여 복합도로 주행

Fig.18 Car Controlled by GA-Improved Rule along Complex Road Alignment

6.3 주행 실험 분석

(그림 15)에서는 자동차가 도로에 수렴하기 위해 커브 브를 뜨는 반면 (그림 16)에서는 개선된 Rule에서 커브를 완만히 하여 핸들링의 횡수가 현저히 줄어든 것을 보여 준다. 또한 차의 절대 이동거리가 (그림 15)와 비교하여 훨씬 줄어든 것을 확인 할 수 있다. 도로중심점과의 거리 면에서는 개선된 Rule이 약간 크기는 하지만 전체적인 주행 코스가 완만해 지고 주행 능력이 향상된 것을 확인해 볼 수 있다.

(그림 17)에서는 자동차가 직선도로에 근접하게 접했을 때에도 불필요하게 핸들링을 하여 직선도로에서도 좌우로 흔들림이 나타나게 되지만 (그림 18)에서 보면 직선 도로에서 도로에 근접했을 때 불필요한 핸들 제어를 하지 않으며, 대각 직선을 만나게 됐을 때에도 흔들림 없이 완만하게 회전하는 것을 확인해 볼 수 있다. 도로의 마지막 부분에서 나타나는 100미터 가량의 회전도로에서는 (그림 17)에서는 잘 적응하지 못하고 주행이 부드럽지 못한 것이 두드러지게 나타나지만 (그림 18)의 경우 매우 완만한 회전을 보여주고 있다.

VII. 결 론

퍼지 시스템을 사용하여 자동차 제어를 하기 위해서 규칙과 소속 함수들을 정의해야 한다. 그리고 사용되는 규칙들 중 필요한 부분과 필요치 않는 부분을 구별하기는 쉽지 않다. 또한 가장 적합하다고 생각되는 소속 함수를 구하는 것도 무리이다. 본 논문에서는 이러한 어려움을 가진 규칙과 소속 함수를 구하기 위해서 유전자 알고리즘을 적용하여 실험하였다. 유전자 알고리즘을 사용하

면 문제에 대한 지식 없이도 제어를 설계할 수 있었으며, 문제에 대한 지식이 있다면 그 지식을 이용하여 신속하면서도 전문가가 정의한 규칙보다 더 좋은 결과를 도출할 수 있었다. 본 실험의 결과에서 알 수 있듯이 유전자 알고리즘은 퍼지 자동차 제어에 필요한 적합한 규칙과 소속 함수를 찾아 줌으로 효과적이고 유연한 자동차 제어를 수행할 수 있도록 했다.

참고문헌

- [1] 장경익, 박성진, 김명원, "진화 알고리즘을 이용한 퍼지 논리 제어기의 최적화" 99년 인공지능 퍼지 시스템학회 종합 학술대회, 1999
- [2] Chuen Chien Lee, "Fuzzy Logic Control Systems: Fuzzy Logic Controller-Part I", IEEE Transactions on Systems, Man, and Cybernetics, vol. 20, No. 2, 1990
- [3] Shin-ichi Horikawa et al., "A Fuzzy Controller Using a Neural Network and Its Capability to Learn Control Rules," Proc. of the International Conference on Fuzzy Logic & Neural Networks, pp.103-106, 1990.
- [4] Hasegawa T., Horikawa S.-i. Furuhashi T., Uchikawa Y., "On design of adaptive fuzzy controller using fuzzy neural networks and a description of its dynamical behavior", Fuzzy Sets And Systems, Vol. 71, Issue 1, pp.3-23, 1995
- [5] F. Hoffmann, et al., Evolutionary Learning of Mobile Robot Behaviours, First Workshop on Frontiers in Evolutionary Algorithms, 1997
- [6] Leonid REZNIK, "Evolution of Fuzzy Controller Design", FUZZ IEEE, 1997
- [7] Tomonori HASHIYAMA, Takeshi FURUHASHI, Yoshiki UCHIKAWA, "A Study on Finding Fuzzy Rules for Semi-Active Suspension Controllers with Genetic Algorithm", First Online Workshop on Evolutionary Computation, 1995
- [8] Frank Hoffmann, Gerd Pfister, "Evolutionary Design of a Fuzzy Knowledge Base for a Mobile Robot", International Journal of Approximate Reasoning ,vol. 17, no. 4, pp.447-469, (1997)

- [9] 윤철민, "유전자 알고리즘을 이용한 제어를 위한 퍼지 규칙 생성", 송실대학교 대학원 전자계산학과 석사 학위 논문, 1998
- [10] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Third, Extended Edition, Springer-Verlag, 1995
- [11] Potter M. A., De Jong K., "A Cooperative Coevolutionary Approach to Function Optimization", Parallel Problem Solving from Nature-PPSN III, pp.249-257, 1994

저자소개

김 봉 기(Bong-Gi Kim)



1987년 송실대학교 전자계산학과
공학사

1989년 송실대학교 전자계산학과
공학석사

1999년 송실대학교 전자계산학과
공학박사

1994년 3월 ~ 1999년 2월 한림성심대학 컴퓨터응용과
조교수

1999년 3월 ~ 현재 진주산업대학교 컴퓨터공학부 부교수
※ 관심분야 지능형 휴, 지능형 제어, 멀티미디어 시스템